PROYECTO FIN DE CARRERA

ING. TÉCNICA DE TELECOMUNICACIONES ESPECIALIDAD SONIDO E IMAGEN

ESTUDIO DE LOS ERRORES DE PALABRA PRODUCIDOS POR EL SERVICIO DE SUBTITULADO AUTOMÁTICO DE APEINTA

Autor: Daniel Cruz Krause

Director: Javier Jiménez Dorado

Tutor: Ana Iglesias Maqueda

Título: ESTUDIO DE LOS ERRORES DE PALABRA PRODUCIDOS POR EL SERVICIO DE SUBTITULADO AUTOMÁTICO DE APEINTA

Autor: Daniel Cruz Krause

Director: Javier Jiménez Dorado

Tutor: Ana Iglesias Maqueda

EL TRIBUNAL

Presidente: Belén Ruiz Mezcua

Vocal: Mercedes de Castro Álvarez

Secretario: Paloma Martínez Fernández

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 24 de Febrero de 2011 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

A mis padres.

A mi hermana.

Agradecimientos

Agradecer a Javi por haberme guiado en este proyecto, pero sobre todo, reconocer el apoyo y la confianza que me ha dado en todo momento. Gracias.

A Ana por la supervisión de este trabajo.

A todos aquellos que de manera directa o indirecta han contribuido a sacar adelante este proyecto.

A mis amigos, por absolutamente todo.

Resumen

Dado que la tecnología pretende solucionar los problemas a los que nos enfrentamos, ésta debe ser la principal herramienta de investigación en el área de la educación. La educación es un derecho inalienable; sin embargo existen innumerables barreras que impiden el acceso a la educación a ciertos colectivos de la población. Uno de los colectivos más perjudicado es de las personas con discapacidad auditiva en edad escolar o universitaria, en pleno desarrollo intelectual. Según datos de la Encuesta sobre discapacidades realizada por el Instituto Nacional de Estadística (INE) en 2008, el número de personas con discapacidad auditiva es de 1.064.200 [1], el 4% de la población española. Estos datos y la perspectiva de futuro obligan a promover un cambio del modelo educativo tal y como lo conocemos hoy en día.

Para hacer frente a los problemas de accesibilidad en entornos educativos, el Centro Español de Subtitulado y Audiodescripción, a través del proyecto **APEINTA** [2]**,** desarrolló un sistema de subtitulado automático, en directo y diferido, que utiliza la tecnología del **Reconocimiento Automático del Habla (RAH)** para transcribir la voz a texto, de modo que los alumnos con discapacidad auditiva pudieran seguir el curso de las clases sin dificultad.

Tras varias evaluaciones de APEINTA se detectó que una de las principales cuestiones a las que se enfrenta el RAH son los errores cometidos por un incorrecto reconocimiento de la voz; estos **errores, denominados en su término anglosajón Word Error Rate (WER), o ratio de error de palabra,** repercuten en la comprensión de las transcripciones y en la pérdida de información para el alumno.

La primera parte del proyecto que se desarrolla en este documento estudia la influencia **del WER en el servicio de subtitulado en diferido de APEINTA, donde automáticamente se permite subtitular vídeos**. Así mismo, también se estudia la **influencia del reentrenamiento** en los sistemas de reconocimiento de voz.

Las transcripciones generadas por sistemas de subtitulado automático y que son utilizadas en el reentrenamiento, deben ser corregidas para aumentar la eficiencia del mismo. Sin embargo, la corrección de los errores de las transcripciones consume mucho tiempo y recursos, por lo que es un aspecto crítico en el proceso global.

La segunda parte del proyecto se centra en encontrar soluciones que asistan en esta tarea y, para ello, se tratará de **modificar el sistema reconocedor para obtener información adicional que apoye y facilite la corrección de las transcripciones**. **En último lugar** se ha construido **un prototipo de editor** de textos que utiliza las nuevas funcionalidades incorporadas.

Abstract

Because the technology is designed to solve the problems we face, this should be the main paradigm of research in the area of education. Education is an inalienable right; however there are numerous barriers that impede access to education to certain groups of population. Hearing impairment people at school or university are one of the groups most affected, due they are in the beginning or in the middle of their intellectual development. According to the Survey on Disability conducted by the National Statistics Institute (INE) in 2008, the number of people with hearing impairment is 1,064,200 [1], 4% of the Spanish population. These data and the prospect of future force to promote a change of the educational model as we know it today.

To face accessibility issues in educational enviroment, the Spanish Center for Subtitles and Audio Description (CESyA), developed a system for automatic subtitling, live and recorded, using the technology of Automatic Speech Recognition (RAH) to transcribe voice into text, named APEINTA project [2]; so that hearing impaired students could follow the classes without difficulty.

After several evaluations APEINTA detected that one of the main problems facing the RAH are the mistakes of an incorrect speech recognition. These errors called Word Error Rate (WER), impact on the understanding of the transcripts and could cause data loss for the student.

The first part of a project developed in this paper examines the impact of WER in the subtitling deferred service of APEINTA, which allows subtitling videos automatically. Likewise, it also examines the impact of enrollment in voice recognition systems.

The transcripts generated by automatic captioning systems that are used for enrollment, should be corrected to increase efficiency. However, the correction of the transcriptions is a time and resource consuming, so it is critical for the global process.

The second part of the project focuses on finding solutions to assist in this task, so we will try to modify the recognition system to get additional information to assist transcript's correction. Finally a text editor prototype that uses the new features was built.

Índice de contenidos

Índice de figuras

1. INTRODUCCIÓN

Este proyecto se ha realizado dentro del marco de investigación del Centro Español de Subtitulado y Audiodescripción (CESyA)[1] que, a través del proyecto **APEINTA,** ha desarrollado un sistema de subtitulado automático en directo y diferido que utiliza la tecnología del **reconocimiento automático del habla** (RAH) para transcribir la voz a texto, de modo que los alumnos con discapacidad auditiva pudieran seguir el curso de las clases sin dificultad.

Con el fin de comprobar las funcionalidades y las características que puede ofrecer el reconocimiento automático del habla en la educación, el proyecto APEINTA se ha evaluado en diferentes escenarios. Estas pruebas, junto con la literatura, demuestran que el reconocimiento automático del habla y por consiguiente APEINTA, pueden mejorar la accesibilidad en las aulas. Sin embargo, en evaluaciones de APEINTA realizadas en años anteriores [3] como las realizadas en este proyecto revelan características del sistema de reconocimiento que reducen su eficacia. Estos reportes indican problemas que dificultan la comprensión de los textos transcritos, debido a los **errores de transcripción** provocados durante el reconocimiento automático del habla.

Dado que este problema no es nuevo en el ámbito del RAH, los sistemas de reconocimiento implementan sistemas para combatir estos errores y mejorar la precisión del motor de reconocimiento. Una de estas herramientas es el **reentrenamiento** del sistema para adaptar los modelos que rigen el reconocimiento a las características de la voz de cada locutor. El reentrenamiento puede ser directo o indirecto. Mientras que en el primero el locutor tiene que estar presente para la lectura de textos de los que se debe disponer de su transcripción previamente, en el reentrenamiento indirecto el audio grabado de las propias sesiones y sus transcripciones libres de errores son las que se utilizan para reentrenar, evitando que el locutor tenga que estar presente durante el reentrenamiento.

Teniendo en cuenta la importancia de estas herramientas para mejorar la precisión de los sistemas de RAH, la **primera parte de este proyecto** estudia, mediante la **evaluación del WER en el subtitulado automático en diferido de vídeos**, la **influencia del reentrenamiento** en los sistemas de reconocimiento de voz.

Para realizar las diferentes pruebas se ha hecho uso del material audiovisual grabado y cedido por las asignaturas de Catalogación de Materiales Especiales y Automatización de Centros y Servicios de Información[2] de la diplomatura de Biblioteconomía y Documentación de la Universidad Carlos III de Madrid[3]. Así mismo, las herramientas utilizadas para medir la tasa de errores de cada vídeo son las provistas

[1] www.cesya.es

[2] Los videos de estas asignaturas son de uso privado, únicamente para los alumnos de estas asignaturas.

[3] www.uc3m.es

por el National Institute of Standards and Technology (NIST[4]) a través del **Speech Recognition Scoring Toolkit (SCTK)** [4] para Linux. Del mismo modo, para llevar a cabo el reconocimiento automático y el reentrenamiento se ha utilizado **Dragon NaturallySpeaking, que es el reconocedor de voz que actualmente se utiliza en APEINTA**.

La **segunda parte del proyecto** es consecuencia de los resultados obtenidos en la primera parte del mismo, que respaldan la eficacia del reentrenamiento. A pesar de la eficacia de esta técnica, se ha detectado que la **corrección de los errores** de las transcripciones generadas por los sistemas de subtitulado automático, necesarias para el reentrenamiento, es un aspecto crítico en el proceso global y la metodología utilizada es subóptima. Es por esto que la segunda parte del proyecto se centra en encontrar soluciones que asistan en esta tarea. Para ello, se **tratará de modificar el sistema reconocedor para obtener información adicional que apoye y facilite la corrección de las transcripciones**. Para incorporar nuevas funcionalidades que asistan a esta tarea se ha modificado el programa APEINTAdragon, encargado del reconocimiento y la transcripción. Por último y con el fin de probar la utilidad de la información complementaria a la transcripción en la corrección de errores se ha construido **un prototipo de editor** de textos.

Para introducir este proyecto esta memoria contiene un estado del arte (Capítulo 2) donde se recogen destacados proyectos. De esta forma, se incluyen los proyectos del consorcio Liberated Learning Project, todos ellos relacionados con las tecnologías de voz, como ViaScribe o Synote; y otros proyectos liderados por el Instituto Tecnológico de Massachusetts (MIT) a través de Spoken Language Systems.

Una vez visto el estado del arte, en el Capítulo 3 repasamos algunos de los trabajos anteriores realizados dentro del proyecto APEINTA que nos permitirá contextualizar los trabajos realizados en este proyecto fin de carrera. En el Capítulo 4 se presentan los objetivos que se marcaron en este proyecto mientras que en el Capítulo 5 se describen en detalle los trabajos realizados, divididos en las dos partes mencionadas anteriormente (Capítulos 5.1 y 5.2). Las pruebas y resultados se explican en el capítulo 6. Para finalizar, se proporciona el presupuesto aproximado del proyecto (Capítulo 7), las conclusiones obtenidas tras analizar los resultados (Capítulo 8) y se reflexiona sobre los posibles trabajos futuros (Capítulo 9). El glosario de términos y la bibliografía consultada más relevante se encuentran en los capítulos 10 y 11 respectivamente.

[4] http://www.nist.gov/index.html

2. ESTADO DEL ARTE

2.1. RECONOCIMIENTO AUTOMÁTICO DEL HABLA

Al igual que el reconocimiento de una señal acústica realizado por el ser humano requiere de un proceso de percepción del sonido, análisis e identificación, el reconocimiento automático del habla (RAH) efectuado por máquinas trata de imitar estas tres fases. Existen diversos paradigmas en el RAH, pero todos ellos buscan transcribir una señal acústica de voz en una cadena de símbolos lo más similar posible.

Dada la naturaleza de la voz el proceso de reconocimiento y transcripción de las palabras y de sus unidades mínimas, los fonemas[5], a su representación escrita, fonogramas[6], es muy complejo.

En primer lugar las limitaciones físicas en el movimiento de un articulador pueden causar variaciones en la pronunciación de los fonemas y dificultar su reconocimiento. La forma acústica de un fonema depende en gran medida de la manera en que se colocan los órganos de articulación (lengua, labios, mandíbula), que hacen que un mismo sonido se pronuncie de diferente manera.

En segundo lugar, los sonidos en el habla no son independientes entre sí, sino que la pronunciación de un fonema determina en gran medida y tiene una gran influencia sobre el fonema que le prosigue. A este efecto se le denomina coarticulación, y es uno de los principales causantes de que en el lenguaje natural la reducción, modificación u omisión de fonemas e incluso sílabas sea uno de los efectos que se presentan con mayor frecuencia.

"It's fun to recognize speech" o "It's fun to wreck a nice beach"

La representación escrita y el significado de estas dos frases difieren mucho entre sí, sin embargo, para un reconocedor de voz es difícil distinguirlas porque son prácticamente iguales acústicamente ya que los fonemas que forman ambas frases tienen unos espectrogramas muy similares. La señal acústica carece de divisiones fácilmente reconocibles entre palabras o fonemas y los efectos causados por los diferentes fenómenos articulatorios, que no representan ninguna dificultad para una persona en una conversación, para un sistema automático de reconocimiento de voz suponen la principal dificultad para transcribir la señal acústica en una cadena de palabras y conseguir una transcripción libre de errores.

En el proceso del reconocimiento de una señal de voz, el sistema trata de encontrar en su diccionario la cadena de símbolos más parecida a los obtenidos como

[5] Fonema: Cada una de las unidades fonológicas mínimas que en el sistema de una lengua pueden oponerse a otras en contraste significativo.

[6] Fonograma: Letra o conjunto de letras que representan un fonema.

señal de entrada. Para ello, un sistema de RAH posee una arquitectura (Fig. 1) con diferentes módulos encargados de extraer y calcular diferentes características y probabilidades de la señal que permitan encontrar la hipótesis que más se aproxima a la frase de referencia (frase expresada por el locutor).

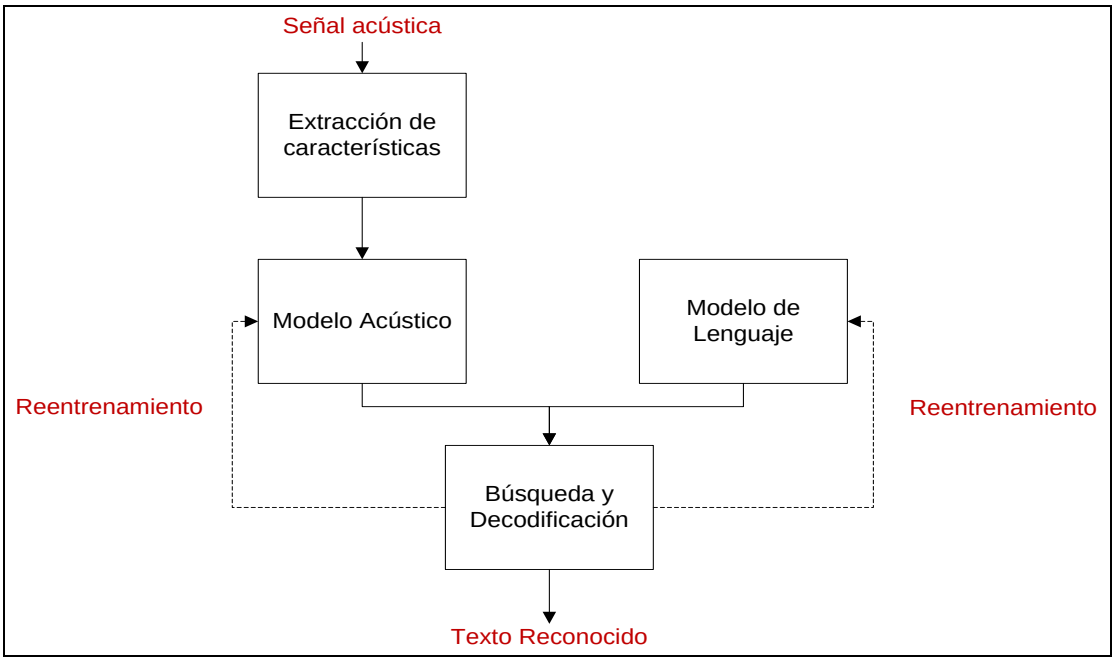


Figura 1. Arquitectura de un sistema de RAH

Durante la **extracción de características**, la señal acústica de entrada se trata como un conjunto de observaciones que se han obtenido al dividir en intervalos de tiempo la señal de entrada y que contienen información espectral y de energía de cada uno de ellos. La señal, por lo tanto, puede expresarse como un conjunto de símbolos denominados observaciones *O,* tales que:

$$O = o_1, o_2, o_3, ..., o_t$$

De la misma forma, una frase se puede representar como un conjunto de palabras *W*:

$$W = w_1, w_2, w_3, ..., w_n$$

Basándose en estas dos tesis se puede decir que la frase hipótesis más parecida a la frase de entrada se calcula como la secuencia de palabras *W* que dada una secuencia de observaciones *O* maximiza la siguiente ecuación:

$$\hat{W} = \arg\max_{W \in L} P(W \mid O)$$

Desarrollando por Bayes[7], la expresión anterior se puede reescribir como:

$$\hat{W} = \arg\max_{W \in L} P(O|W)P(W)$$

Donde $P(O|W)$ es la verosimilitud y $P(W)$ la probabilidad a priori.

Finalmente en la **fase de búsqueda y decodificación**, con las verosimilitudes de una secuencia dada de símbolos calculadas según el **modelo acústico**, las probabilidades a priori del **modelo de lenguaje** y con un diccionario de pronunciación de las palabras se obtiene la secuencia de palabras más parecida a la señal de entrada.

Para que el proceso de reconocimiento sea exitoso, además de tener unos modelos acústicos y de lenguaje muy desarrollados, es necesario el **reentrenamiento** del sistema con el mayor número de horas de voz posibles. Para ello, como se muestra en la Fig. 1, el texto resultante del reconocimiento se utiliza, junto con el archivo de audio que contiene el discurso, para actualizar y mejorar la precisión de los módulos acústicos y lenguaje del sistema.

2.1.1. SISTEMAS DE RAH

Existen diversas herramientas de reconocimiento de la voz en la actualidad, pero aquí se destacarán tan sólo algunas de ellas cuya calidad o relevancia académica son de interés para este proyecto. Los sistemas que se muestran a continuación cumplen aquellos parámetros y funciones que son especialmente útiles dentro del marco del proyecto APEINTA (ver sección 3.1 en Trabajos Anteriores). Estas aplicaciones permiten el reconocimiento de habla continua de gran vocabulario y en tiempo real; ofrecen la posibilidad de adaptar y modificar el software –mediante SDK, API u otros- para diseñar nuevas aplicaciones avanzadas; permiten el reentrenamiento y optimizar el perfil de cada usuario con nuevo vocabulario para mejorar la calidad del reconocimiento. Otros aspectos secundarios, pero importantes, son la variedad de vocabularios, idiomas, compatibilidad con diversos sistemas operativos y dispositivos que pueden soportar estas herramientas.

Dragon NaturallySpeaking (DNS) [5]: Desarrollado por Nuance [6], es uno de los motores de reconocimiento más potentes y con tasas de error muy bajas. Dragon ofrece, además, una interfaz para que el usuario gestione las diferentes opciones del reconocimiento, herramientas que facilitan el entrenamiento de los perfiles y la

[7] $\hat{W} = \arg\max_{W \in L} \frac{P(O|W)P(W)}{P(O)}$ Dado que se maximiza sobre todas las posibles frases, la expresión se calcula para cada frase, teniendo en cuenta que $P(O)$ no varía.

visualización y corrección de las transcripciones. Nuance también comercializa un SDK y un API para el desarrollo de aplicaciones que utilicen su motor de reconocimiento de habla.

ViaVoice (VV) [7]: Desarrollado por IBM desde 1997, es un sistema de RAH similar a DNS por su capacidad de transcripción de habla continua y de gran vocabulario. Sin embargo, desde 2005, año en el que Nuance –propietaria de DNS- comprara los derechos de ViaVoice, no se da soporte a este producto. De este modo, la versión actual de IBM Embedded ViaVoice funciona únicamente en sistemas integrados, como sistemas de control de automóviles o dispositivos móviles [8]. Este software se caracteriza por ser el motor de reconocimiento de sistemas de subtitulado automático como ViaScribe, que se explicará en el siguiente capítulo.

Otros motores de reconocimiento comerciales que se pueden encontrar hoy en día relacionados con el reconocimiento del habla son Loquendo[8], con una amplia gama de soluciones de RAH para el mercado de la telefonía, o VERBIO[9], empresa española especializada en ofrecer tecnología de RAH en procesos industriales o logísticos, en entornos domóticos y en productos de soporte en el segmento de la discapacidad.

Los diferentes programas mencionados hasta el momento se caracterizan por ser productos que incorporan tanto tecnología de RAH como interfaces para que el usuario pueda hacer uso de la misma en las diferentes aplicaciones. Sin embargo, existen otras soluciones centradas en proveer las herramientas necesarias para construir desde cero aplicaciones con tecnología del habla o incorporar el reconocimiento de la voz a otras ya existentes. Además, las aplicaciones que se mencionan a continuación permiten por su condición de software libre acceder y modificar la totalidad del código y sin coste alguno.

Hidden Markov Model Toolkit (HTK) [9]: Este motor de reconocimiento ha sido desarrollado en el departamento de ingeniería de la Universidad de Cambridge (CUED) [10]. HTK es un conjunto de herramientas que permiten la construcción y manipulación de los Modelos Ocultos de Markov (MOM). Estos modelos están diseñados para poder realizar el reconocimiento de habla continua o aislada y de gran vocabulario. Una de las características más atractivas de este software es que posibilita la construcción de aplicaciones personalizadas haciendo uso de las librerías (C++) que ofrece HTK bajo el nombre de ATK (Real-Time API para HTK). Esta herramienta es gratuita y de código libre.

CMUSphinx [11]: CMU Sphinx Toolkit, al igual que HTK, es un conjunto de herramientas de reconocimiento automático de voz desarrollado por la Universidad Carnegie Mellon [12]. CMU Sphinx está diseñado para ofrecer soluciones prácticamente en cualquier entorno que requiera reconocimiento automático del habla. En su última

[8] http://www.loquendo.com/es/
[9] http://www.verbio.com/webverbio3/html/index.php

versión (Sphinx-4), la plataforma de programación es Java[10], mientras que en las versiones anteriores es Visual C/C++. Esta herramienta es gratuita y de código libre.

Existen otros motores de reconocimiento similares a HTK y Sphinx, de licencia gratuita y de código libre como JULIUS[11], desarrollado en el Kawahara Lab., de la Kyoto University[12]. Éste, al igual que HTK y Sphinx, es un decodificador de voz que soporta habla continua de gran vocabulario entre otros. Como el HTK, utiliza Modelos Ocultos de Markov para la construcción del modelo acústico y es capaz de realizar el reconocimiento del habla –dictado y palabras aisladas- en tiempo real usando vocabularios de hasta 60 mil palabras. Una característica de este software es que adopta formatos estándares para asegurar la compatibilidad con otras herramientas de reconocimiento como HTK. La principal plataforma de JULIUS es Linux y otras distribuciones Unix, aunque también funciona en Windows.

Aunque no se ha encontrado una comparativa de los tres últimos decodificadores de voz mencionados, sí existe en la literatura una evaluación de HTK y CMUSphinx que muestra que ambos decodificadores ofrecen unos niveles de WER similares, en torno al 20% usando un vocabulario de 60 mil palabras (60k), y de un 6% para un vocabulario de 5 mil palabras (5k) [13]. Aunque HTK es más complejo, es más flexible y la documentación ofrecida para construir aplicaciones basadas en el reconocimiento de voz en diferido o en tiempo real es mejor que la ofrecida para CMUSphinx. Sin embargo, una de las ventajas de CMUSphinx frente a HTK es que puede incluirse en aplicaciones comerciales sin ningún tipo de limitación.

2.1.2. PROYECTOS BASADOS EN RAH PARA LA ACCESIBILIDAD

Existen diversos proyectos que utilizan el reconocimiento de voz para llevar la accesibilidad a diversos escenarios que hasta el momento las personas con discapacidad auditiva tenían dificultades para acceder y disfrutar. Dos de los campos más importantes donde se están haciendo esfuerzos para salvar las barreras son el ocio y la educación. En el primer caso, algunos sitios de internet como Youtube.com[13], portal de vídeos por streaming, ha incorporado el RAH y ofrece subtitulado automático y sincronizado de sus vídeos. A través de este sistema se permite que colectivos de personas con discapacidad auditiva, que no podían disfrutar de este portal, puedan acceder a toda la información audiovisual que ahora se les brinda.

Aunque existen tecnologías de reconocimiento de voz y subtitulado automático como la de Google, son muy pocas las herramientas de subtitulado automático enfocadas a la educación.

[10] http://cmusphinx.sourceforge.net/sphinx4/
[11] http://julius.sourceforge.jp/en_index.php?q=index-en.html#about_julius
[12] http://www.ar.media.kyoto-u.ac.jp/
[13] http://www.youtube.com

Liberated Learning Project

IBM ViaScribe es la aplicación más importante de subtitulado en tiempo real enfocado al campo de la educación. Esta herramienta ha sido diseñada para ofrecer en un mismo espacio múltiples fuentes de información multimedia sincronizada. El hablante puede crear una presentación multimedia que aúna el audio de su discurso, subtítulos accesibles (*Text Display Panel*), diapositivas, anotaciones y vídeos (*Tabbed Pane*) (Fig. 2). El resultado puede ser editado, corrigiendo la transcripción del audio o editando el vídeo, para mejorar la calidad de la información brindada.

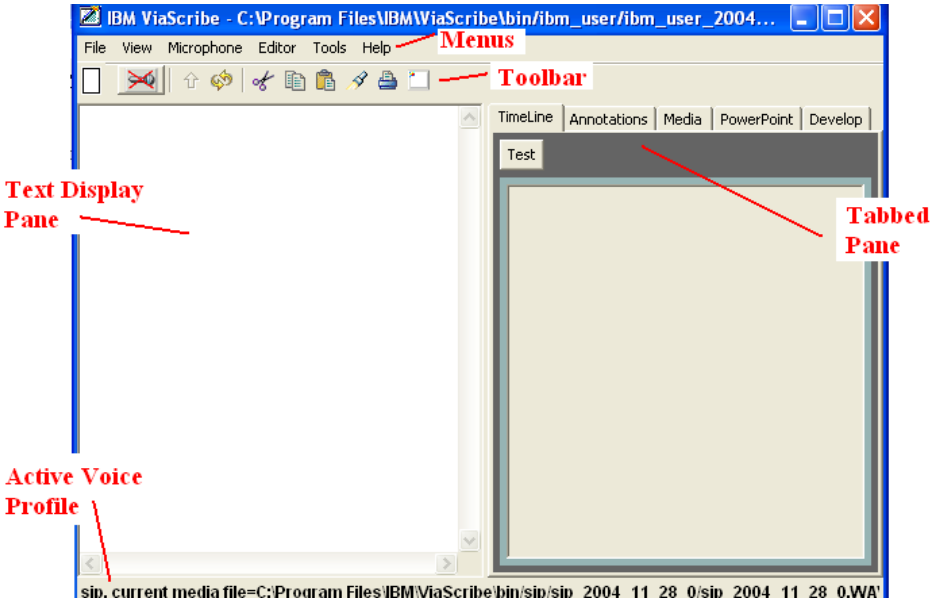


Figura 2. Interfaz de IBM ViaScribe.

ViaScribe utiliza ViaVoice como motor de reconocimiento automático de voz, siendo necesario elegir un perfil de voz asociado al locutor para mejorar la calidad del reconocimiento (*Active Voice Profile*). Una ventaja de este reconocedor es que, a diferencia de Dragon, el motor de reconocimiento del habla introduce un menor retardo entre que se pronuncia una palabra y ésta es transcrita (el Capítulo 3.2 analiza esta cuestión de una forma más profunda).

En situaciones donde existe más de una persona hablando en una misma clase, utilizando varias instancias de ViaScribe y la aplicación RealTimeMerge[14], se pueden mostrar a la vez y en tiempo real las transcripciones de cada uno de los streams; éstos están etiquetados para identificar al locutor.

Como alternativa a ViaScribe desde 2008, se encuentra Synote [14], una aplicación web gratuita que permite crear anotaciones sincronizadas con clips de audio, vídeo, transcripciones de los mismos, diapositivas e imágenes (Fig. 3). Esta herramienta, desarrollada en la Universidad de Southampton, a diferencia de ViaScribe no muestra en tiempo real la transcripción de la voz, sino que sincroniza todo el material a posteriori.

[14]Klaus Miesenberger, "Computers helping people with special needs". Pág. 619 y 620.

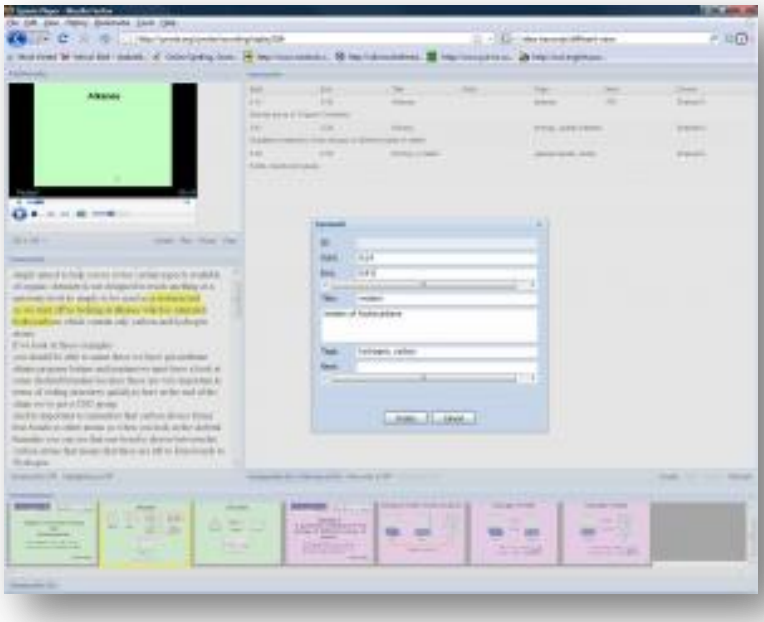


Figura 3. Interfaz de Synote

Uno de los principales atractivos de Synote es la transcripción de los archivos de audio y vídeo en remoto, vía web –si se utiliza en la Universidad de Southampton-, sin la necesidad de que el usuario tenga ningún programa de RAH. Se está trabajando para ofrecer el mismo servicio con Dragon NaturallySpeaking, sin embargo, actualmente sólo se ofrece la posibilidad de utilizar transcripciones ya realizadas en local con aplicaciones como Dragon Naturally Speaking o la aplicación de RAH de Microsoft.

Massachusetts Institute of Technology: Spoken Lecture Processing Project

Un proyecto similar al del consorcio Liberated Learning es el desarrollado por el Instituto Tecnológico de Massachusetts [15], Estados Unidos. Esta universidad, a través del Laboratorio de Informática e Inteligencia Artificial, ha desarrollado una herramienta de transcripción e indexación automática de archivos de audio procedente de las clases impartidas (*spoken lectures*), junto con un buscador que permita localizar partes específicas de estas clases. Spoken Lecture Processing Server es una herramienta web que en la cual los usuarios pueden subir archivos de audio para que un sistema de reconocimiento de voz alojado en un servidor los transcriba a texto [16]. Para mejorar las prestaciones y precisión del reconocedor de voz se permite adjuntar documentos de texto como apuntes o capítulos de libros, que son usados para adaptar el modelo de lenguaje y el vocabulario del sistema.

Las evaluaciones del WER de 10 transcripciones generadas por este sistema lo sitúan en torno al 40%. Sin embargo, se debe destacar la precisión del sistema a la hora de recuperar los segmentos de audio que contienen las palabras clave que coinciden con la búsqueda realizada, llegando al 90% [16]. Por otra parte, destaca por la excelente discriminación de sonidos que no pertenecen al discurso, debido a que el archivo de

audio es procesado por un detector de habla, previo al reconocimiento propiamente dicho, eliminando todos aquellos segmentos del audio que no se corresponden con habla continua.

Como complemento al sistema de reconocimiento, transcripción e indexación del audio de las clases se ha desarrollado un buscador, llamado Lecture Browser, en el que el usuario realiza una búsqueda y obtiene una lista con los resultados de las clases indexadas que coinciden con ésta (Fig. 4).

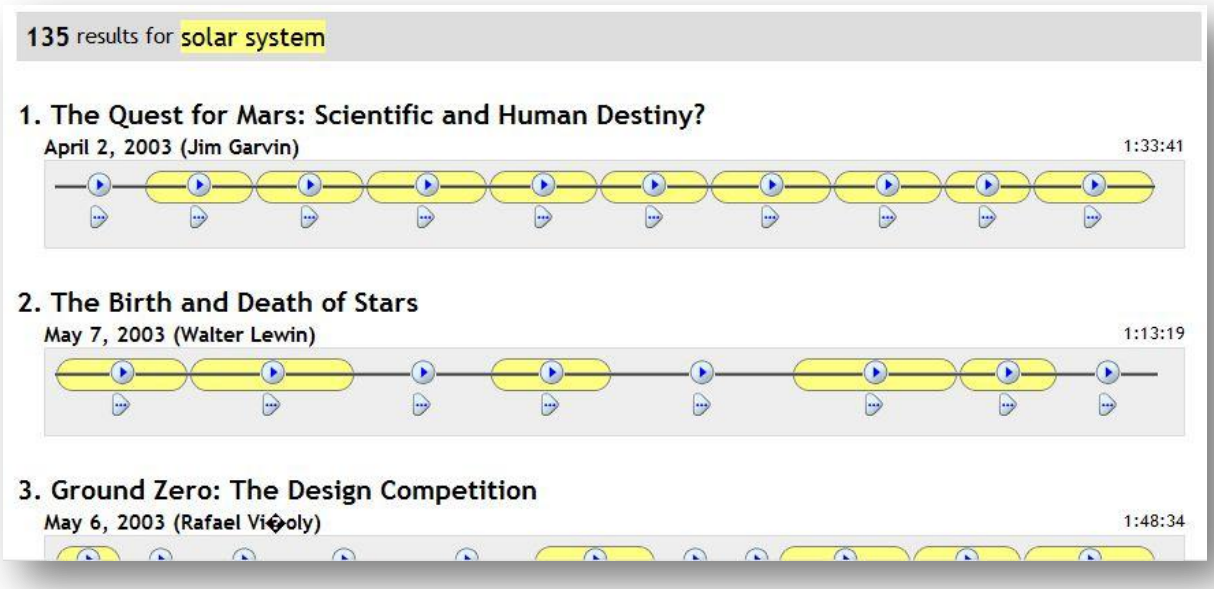


Figura 4. Resultados que coinciden con la búsqueda "solar system" ofrecidos la herramienta Lecture Browser.

Dentro de cada resultado se ofrece el contexto de la transcripción por cada coincidencia con la búsqueda realizada (Fig. 5).

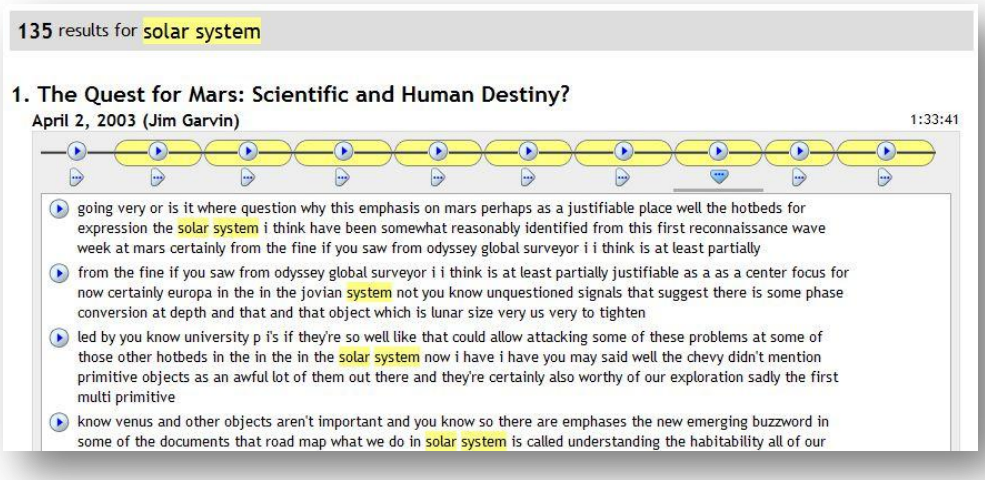


Figura 5. Desglose de los resultados que coinciden con "solar system" dentro de un mismo archivo de audio.

Si se pincha sobre cualquiera de los resultados se mostrará el vídeo y transcripción sincronizados –la palabra subrayada es la que está siendo pronunciada en ese instante- desde el mismo momento en el que éstos fueron pronunciados (Fig. 6).

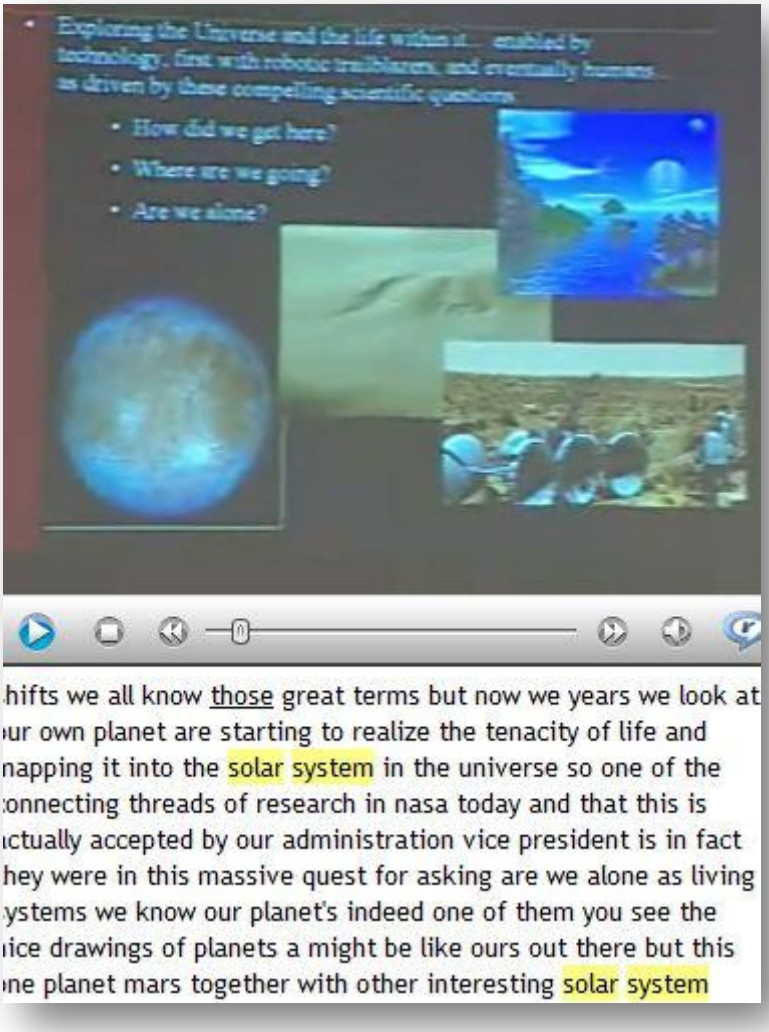


Figura 6. Parte de la interfaz de Browser Lecture en la que se muestra el vídeo y la transcripción sincronizados de uno de los resultados de la búsqueda "solar system".

2.1.3. SISTEMAS DE EDICIÓN ASISTIDA DE SUBTÍTULOS

La corrección del texto resultante del reconocimiento es vital para ofrecer un servicio de calidad y para poder reentrenar los sistemas de RAH para mejorar la precisión de los mismos. Para facilitar la corrección del texto y que ésta pueda hacerse en tiempo real o para reducir los tiempos de corrección a posteriori, es necesario utilizar herramientas de corrección más complejas, y específicas para el reconocimiento de voz. Hoy en día hay una gran escasez de herramientas para tal efecto, limitándose a sistemas muy rudimentarios como bloc de notas o editores de subtítulos. Sin embargo,

existen otras que, pensadas para estos casos, sí facilitan y reducen los tiempos de corrección, permitiendo incluso la edición de los textos en tiempo real.

Es el caso del sistema desarrollado por IBM, que puede utilizarse junto con ViaVoice o Dragon NaturallySpeaking 9, conocido como Caption Editing System (CES) [17]. Una de las principales ventajas de este sistema es su sencilla interfaz, que puede ser utilizada sin apenas entrenamiento acerca del manejo del sistema, y la posibilidad de que la corrección se lleve a cabo por varias personas a la vez –éstas se encuentran conectadas en línea de forma que pueden trabajar de forma simultánea sobre un mismo texto. Existe un editor, denominado *Master Editor,* puede realizar la corrección o bien distribuir esta tarea entre los diferentes editores clientes (*Client Editor*). Toda la corrección puede realizarse sin ayuda del ratón, tan solo con el teclado, lo que reduce el tiempo de edición. En la Fig. 7 y Fig. 8 se muestra la interfaz del CES para las versiones Master y Client.

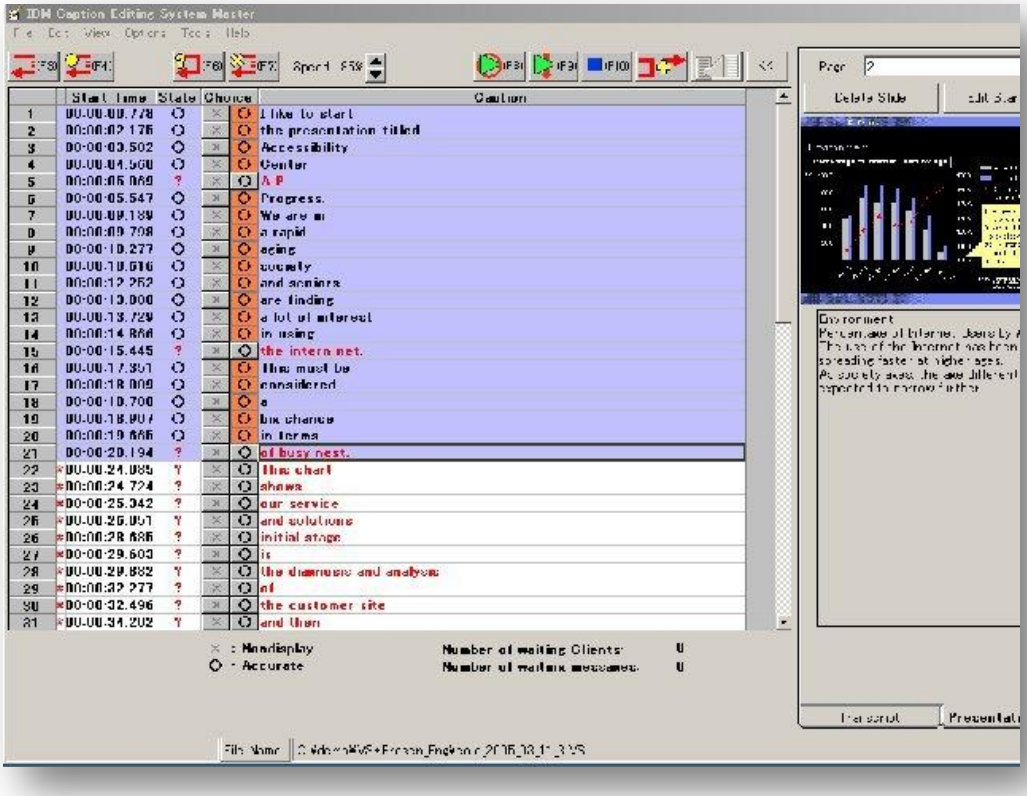


Figura 7. Master Editor del CES

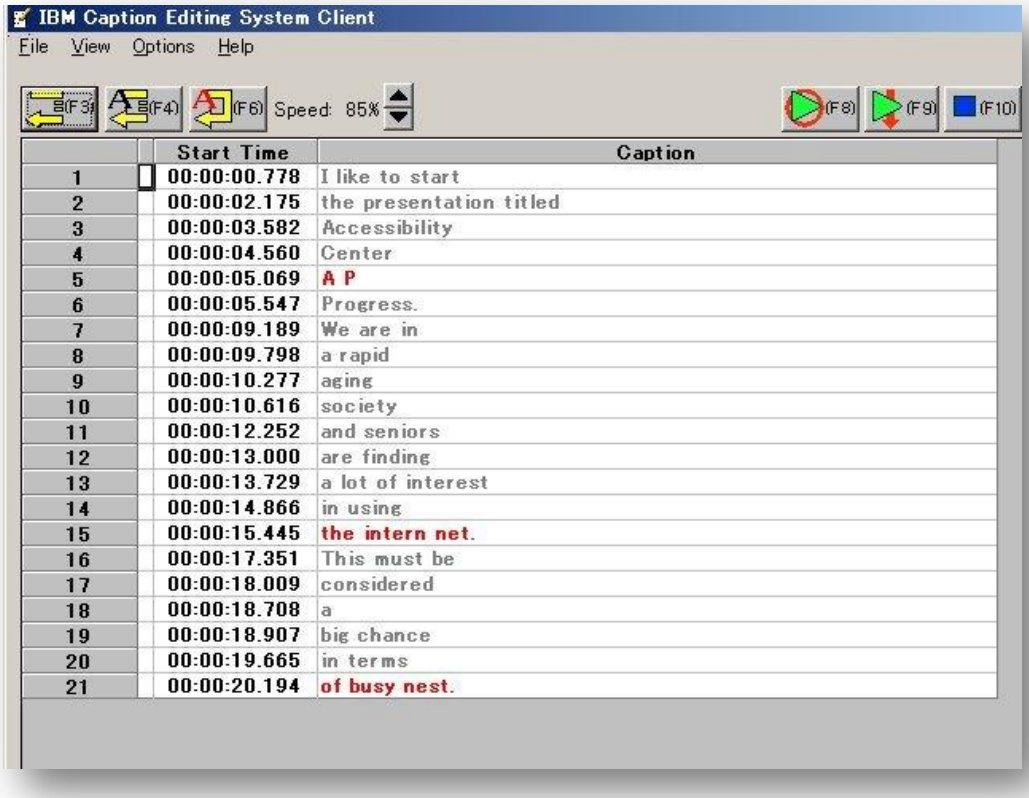


Figura 8. Client Editor del CES

Otros sistemas de reconocimiento de voz como Dragon, ViaVoice o aplicaciones como ViaScribe o Synote, no incorporan herramientas propias de corrección optimizadas para la reducir los tiempos de edición del texto transcrito.

2.1.4. ESCENARIOS DE APLICACIÓN DEL RAH

Las aplicaciones de reconocimiento del habla se encuentran por una extensa variedad de áreas. Los avances en este campo han permitido crear herramientas a la medida de cada industria y por tanto los sistemas de reconocimiento están cada vez más extendidos.

En medicina el reconocimiento de voz se utiliza para controlar dispositivos, lo que permite realizar tareas *hands-free* o, en combinación con herramientas telemáticas médicas, como las historias clínicas electrónicas (HCE), ha permitido mejorar y hacer más rápidas y flexibles ciertas tareas médicas.

Por otro lado, el sector fabril ha sido una de las industrias donde antes se acogieran los avances en reconocimiento de voz, formando parte de los procesos de control, líneas de producción o diseño. Otros sectores, como el de la banca, también han sabido aprovechar las ventajas de las aplicaciones basadas en la voz. Se han desarrollado herramientas que permitían conocer información financiera en tiempo

real. Algunos bancos como Caja Madrid, BBVA o Banesto, entre otros muchos, permiten realizar movimientos bancarios y gestionar productos financieros con la tecnología del reconocimiento del habla de NaturalVox[15] y sin intervención humana.

También, los nuevos dispositivos, como teléfonos móviles, PDAs o tablets, incorporan sistemas de reconocimiento de voz, nativos o bien con aplicaciones de terceros, para realizar determinadas tareas, como el marcado de dígitos por voz, navegación por la interfaz del dispositivo o inclusive el dictado de notas y documentos de texto –Dragon NaturallySpeaking para dispositivos móviles[16].

Dada la naturaleza de este proyecto nos centraremos en la penetración de la tecnología del RAH en la educación, y cómo ésta actúa como vehículo para hacer accesible aspectos de la enseña que hasta ahora eran de difícil acceso para personas con algún grado de discapacidad. Un ejemplo, junto con los teclados Braille, son los sistemas de reconocimiento de voz utilizados para controlar el ordenador. Estos han significado una alternativa a los métodos tradicionales como el ratón y el teclado, que hacían muy difícil su uso y han permitido acceder a todos los contenidos que pueden ofrecer estos dispositivos al colectivo de personas con discapacidad visual o física.

Algunos de los problemas con los que se encuentran son de comprensión del lenguaje o la inteligibilidad del habla, aunque pueden existir dificultades personales añadidas que hacen más compleja su situación. Estos hechos obligan a promover un cambio del modelo educativo tal y como lo conocemos hoy en día, y esta evolución pasa por promover cambios en la respuesta educativa al alumnado con dificultades auditivas para llegar a eliminar las barreras de comunicación.

Para dar respuesta a estas exigencias se ofrecen soluciones de muy distinta índole [18]: desde la presencia de signantes en los centros educativos, ayudas técnicas individuales y colectivas hasta la incorporación de nuevas tecnologías para el apoyo a la enseñanza –como los sistemas ViaScribe o Synote entre otras soluciones. En esta última área destaca la labor y la tecnología desarrollada por el Centro Español de Subtitulado y Audiodescripción (CESyA), a través del proyecto APEINTA, para favorecer la integración de este colectivo en la educación.

[15] http://www.natvox.es/es/index.aspx
[16] http://www.dragonmobileapps.com/

3. TRABAJOS ANTERIORES

En este capítulo se describen trabajos previos a este proyecto y que han servido como referencia para su realización. De este modo, se presenta por un lado el proyecto APEINTA y, en segundo lugar, la evaluación de su servicio de subtitulado automático en diferentes escenarios.

3.1. APEINTA, UN PROYECTO QUE APUESTA POR LA ENSEÑANZA INCLUSIVA DENTRO Y FUERA DEL AULA

El proyecto APEINTA, desarrollado por el Centro Español de Subtitulado y Audiodescripción (CESyA) junto con la Universidad Carlos III de Madrid, promueve la accesibilidad a la enseñanza a través de la educación inclusiva mediante el uso de nuevas tecnologías tanto informáticas como telemáticas. Este proyecto se desarrolla en dos escenarios distintos: dentro del aula y fuera de ella.

En primer lugar, APEINTA trata de eliminar las barreras de comunicación que existen dentro del aula. Para ello (Fig. 9), apuesta por el uso de sistemas de reconocimiento del habla (RAH) para proporcionar transcripción en tiempo real y en diferido en forma de subtítulos o texto plano, además de mecanismos de síntesis de voz (TTS, Text-To-Speech) como apoyo a la comunicación entre profesor y alumno. El segundo escenario de aplicación de este proyecto se sitúa fuera del aula, donde se ofrece una plataforma accesible de enseñanza Web con una gran variedad de recursos educativos (RE).

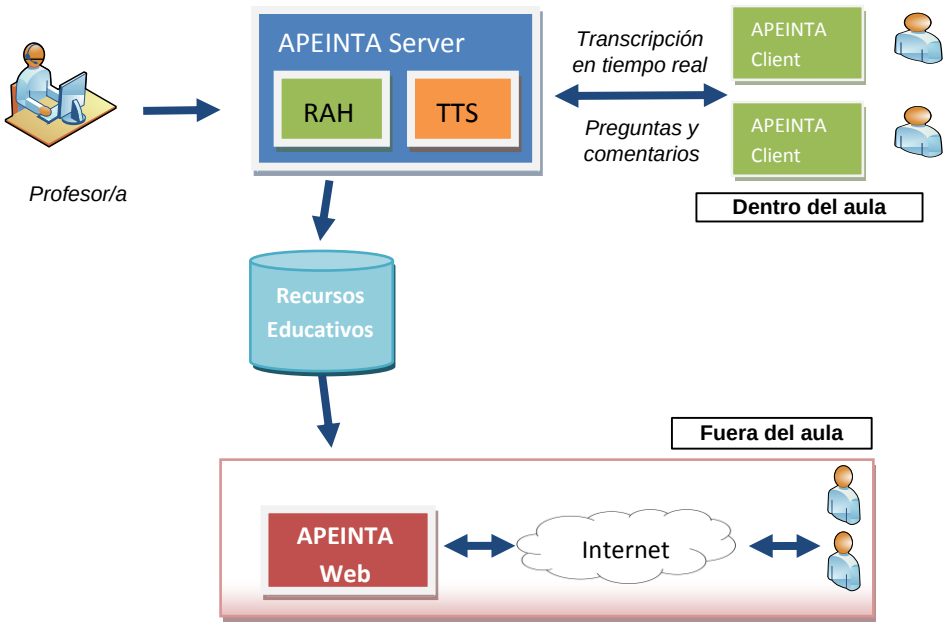


Figura 9. Arquitectura de APEINTA

La arquitectura del sistema está diseñada como una plataforma cliente/servidor en la que se diferencian diferentes servicios con un funcionamiento multidispositivo. El servidor, desarrollado en LabVIEW v.8.5[17], se encuentra en el ordenador del profesor y tiene como función principal la transcripción de la señal de voz y el envío del texto a las diferentes unidades clientes –ubicadas en los diferentes dispositivos que tienen los estudiantes.

En la actualidad APEINTA Server utiliza el sistema de RAH Dragon NaturallySpeaking (DNS), capaz de transcribir habla continua y de gran vocabulario. Para implementar el motor de RAH en la aplicación de APEINTA Server se utilizó el Software Development Kit v.9 (SDK) de DNS, programando en Visual C++[18] una aplicación diseñada para establecer comunicaciones mediante conexiones TCP[19] entre el servidor y el cliente.

Por otro lado, el servidor se encarga de recibir los mensajes de texto escritos por los estudiantes en los dispositivos cliente y transformarlos en una voz sintética por medio del módulo de *Text To Speech* (TTS). Para desarrollar esta aplicación se utilizó Speech Application Programming (SAPI[20]) v.5 de Microsoft. Esta aplicación permite modificar el tono de voz que se utilizará, así como la velocidad y el volumen de ésta.

El funcionamiento de APEINTA dentro del aula y en un escenario de subtitulado en tiempo real, como se describe en la Fig. 9, es el siguiente. El módulo llamado APEINTA Server se utiliza para generar en tiempo real recursos educativos como subtítulos sincronizados o notas en diferentes formatos, que son mostrados al estudiante y que se generan a partir de la transcripción de la señal de voz del profesor/a. Esta información se muestra al alumno a través de distintos dispositivos como pantallas de televisión o puede ser enviada a través de un servidor web a un ordenador portátil (Fig. 10), a una PDA o teléfono móvil (Fig. 11). Paralelamente, el módulo de generación de voz sintética (TTS) permite que haya una canal de comunicación bidireccional, en el cual los asistentes que lo requieran, por ejemplo estudiantes con problemas de pronunciación, haciendo uso de esta tecnología puedan participar planteando comentarios o dudas.

[17] LabVIEW es un entorno de programación gráfica que permite desarrollar sistemas de medida, pruebas y control utilizando un lenguaje de programación gráfico. [http://www.ni.com/labview/].

[18] Entorno de desarrollo integrado (IDE) para lenguajes de programación C y C++, así como para otras API's de Windows como .NET y DirectX [http://www.microsoft.com/express/Windows/].

[19] Protocolo de control de transmisión que permite establecer una conexión entre dos sistemas para el envío de un flujo de datos.

[20] API de Windows para la programación de aplicaciones que hacen uso de tecnología de voz, como reconocimiento del habla o la síntesis de voz [http://msdn.microsoft.com/en-us/library/ms723627(v=VS.85).aspx]

Figura 10. Interfaz de APEINTAServer en modo subtítulo o texto plano.

Figura 11. Interfaz de APEINTAserver en PDAs.

Además del documento con los subtítulos sincronizados en formato SRT[21], el sistema genera dos archivos adicionales, uno de audio (WAV[22]) con la locución del profesor y un fichero XML[23] con la transcripción de la sesión para utilizarlos como recursos educativos aprovechables *Fuera del aula*. De este modo, todos los recursos son almacenados en una base de datos y son accesibles a través de varios servicios web (APEINTA Web), justificando de este modo, la presencia de APEINTA fuera del aula.

Aunque la función principal del servicio de subtitulado de APEINTA es la de subtitulado automático del habla para directo, también se ha añadido una segunda funcionalidad que permite la transcripción de audio y vídeo grabados previamente (Fig. 12).

[21] Formato de texto utilizado para definir subtitulos

22 WAVeform audio file format, formato de audio digital sin compresión de datos desarrollado por Microsoft.

23 Siglas en inglés de eXtensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por elWorld Wide Web Consortium (W3C).

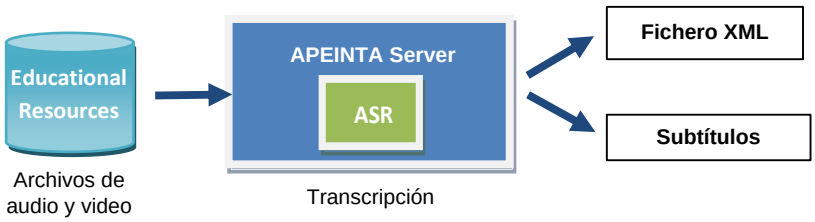


Figura 12. Transcripción de RE pregrabados

Los archivos generados también pueden ser utilizados como recursos docentes y acceder a los mismos a través de los servicios web con los que cuenta APEINTA.

3.2. EVALUACIÓN DEL SISTEMA DE SUBTITULADO AUTOMÁTICO DE APEINTA DENTRO DEL AULA

El proyecto APEINTA ha sido evaluado en tres escenarios diferentes, centrándose en el subsistema que actúa *Dentro del aula* [3].

Estas evaluaciones se realizaron en estudiantes con sordera profunda, estudiantes con dificultades de audición y estudiantes sin ningún tipo de discapacidad auditiva. Los lugares de estudio fueron el Colegio Tras Olivos de Madrid[24], colegio adaptado a personas con discapacidad auditiva, en el 2008; la Universidad Carlos III de Madrid[25], en la asignatura de Diseño de Bases de Datos de Ingeniería Informática en el curso académico 2007-08, y en el congreso ACAPPS[26] (Federación de Familias y Personas Sordas de Cataluña), en 2009.

En el primero de los escenarios, en el Colegio Tres Olivos de Madrid, se realizaron las pruebas sobre 6 niños de entre 10 y 14 años con diferentes tipos de sordera, más un grupo de niños sin ningún tipo de discapacidad auditiva. El sistema se probó con PDAs y ordenadores portátiles, por lo que el resto del aula no veía los subtítulos.

El segundo lugar de evaluación fue la Universidad Carlos III de Madrid, en la asignatura de Diseño de Bases de Datos de Ingeniería Informática durante 50 minutos de duración. En este caso, ninguno de los estudiantes tenía discapacidad auditiva alguna y únicamente 11 de los 46 participantes en la evaluación utilizaban clientes APEINTA (RAH y TTS); además de la pantalla de televisión a la vista de toda el aula que mostraba los subtítulos o texto plano transcritos. La tasa de error a nivel de palabra en la transcripción generada por el sistema de RAH fue del 10,4% [3].

[24] Colegio de educación primaria y secundaria de integración preferente de discapacitados auditivos http://www.colegiotresolivos.org/
[25] www.uc3m.es
[26] http://www.acapps.org/web/

El último escenario de prueba fue en la conferencia de la ACAPPS, donde 5 de los asistentes tenía problemas de audición y usaban dispositivos para recibir los subtítulos generados por APEINTA. El error a nivel de palabra en los subtítulos fue del 20% [3].

En las tres evaluaciones descritas anteriormente tanto los usuarios de APEINTA como los no usuarios tuvieron una respuesta positiva. Las personas con discapacidad auditiva consideraron positivamente el sistema, mientras que para el resto de alumnos, sin discapacidad alguna, el sistema no supuso ningún beneficio ni perjuicio.

Un gran porcentaje de los usuarios notaron un retardo apreciable entre el tiempo de la voz y el de la transcripción, lo que aumentaba la dificultad en la comprensión y el seguimiento del discurso. Una segunda manifestación fue la tasa de error de la transcripción. Sin embargo, a pesar de obtener unos resultados mejorables en términos del WER (Word Error Rate) y el retardo, la tecnología del RAH ha demostrado tras estas evaluaciones ser una buena herramienta para llevar la accesibilidad a la educación.

4. OBJETIVOS

El sistema de subtitulado automático en el entorno educativo desarrollado en el marco del proyecto APEINTA ha demostrado que la tecnología del reconocimiento automático del habla puede mejorar la accesibilidad en las aulas y servir de complemento a otras soluciones más tradicionales como audífonos o sistemas de inducción magnética, que suponen un apoyo indiscutible para mejorar los niveles de accesibilidad para las personas con discapacidad auditiva.

La presencia de sistemas de subtitulado automático puede ser una alternativa en algunas situaciones a la estenotipia o el LSE (Lenguaje de Signos Español). Mientras que la formación de estenotipistas y signantes es costosa en términos económicos y de tiempo y el LSE no es utilizado por todas las personas con discapacidad auditiva, el RAH puede ofrecer una solución eficaz, ubicua, escalable y global mediante la transcripción a texto de cualquier discurso oral.

Algunas de estas razones perfilan a los sistemas de reconocimiento de voz como una alternativa eficaz para mejorar la accesibilidad en la educación.

Sin embargo, las evaluaciones realizadas en los diferentes centros educativos revelan características del sistema de reconocimiento que dificultan la comprensión de la transcripción. La primera de ellas es la *tasa de error* de palabras erróneas (WER) en texto transcrito, fallos cometidos por el reconocedor al identificar los fonemas erróneos a partir del sonido producido. De acuerdo con la literatura, la comprensión de un discurso se puede realizar correctamente siempre que su WER no supere un umbral del 15% de errores [19].

El segundo de los problemas detectados es el *tiempo de reconocimiento (retardo)*, tiempo que transcurre entre la recepción de la señal acústica y el momento en el que se devuelve la transcripción de la misma. Algunos datos medidos en el proyecto APEINTA muestran retardos de hasta 30 segundos, sobre todo en situaciones en las que el orador aumenta la velocidad del discurso y realizaba menos pausas entre frases [3]. El retardo provocado en estas situaciones se debe a que el sistema de RAH en el que se basa DNS espera a que se pronuncie la frase completa para realizar su análisis y reconocimiento. Este proceso supone menores tasas de error puesto que se utiliza el contexto como ayuda para la identificar la mejor transcripción, a cambio de mayores tiempos de retardo[27].

Haciendo uso de esta información y tomando APEINTA como punto de partida, este proyecto consta de dos partes:

- Estudio de la influencia del reentrenamiento en el subtitulado automático para reducir el WER.

[27] Otros sistemas de reconocimiento de voz, como ViaVoice, realizan un reconocimiento palabra por palabra lo que disminuye el retardo –a costa de tasas de error más altas.

- Optimización del proceso de corrección de errores.

Dada la influencia de la tasa de error (WER) en el subtitulado automático y la importancia que este tiene en la comprensión de la transcripción, la primera parte de este proyecto se centra en evaluar el WER de vídeos y estudiar su evolución a medida que el sistema reconocedor es reentrenado por *enrollment* –se trata de un proceso de reentrenamiento indirecto del sistema, sin presencia del orador (más información en el Capítulo 5.1.1). Para realizar la evaluación se ha utilizado material audiovisual obtenido de las asignaturas de Catalogación de Materiales Especiales y Automatización de Centros y Servicios de Información de la diplomatura de Biblioteconomía y Documentación de la Universidad Carlos III de Madrid. A raíz del análisis realizado se ha obtenido un conocimiento que introduce y justifica la segunda parte de este proyecto.

Una de las conclusiones de este primer apartado es que existe una reducción del WER y una mejora de la precisión durante el reconocimiento cuando se realiza un reentrenamiento del sistema por *enrollment*. Ante estos resultados se deduce que el *enrollment* es una buena herramienta que mejoría los resultados en cuanto a precisión se refiere, situándose como una alternativa más eficiente que el subtitulado desde cero.

Sin embargo, este tipo de reentrenamiento requiere de transcripciones literales y libres de errores, lo que implica un proceso de corrección previo de las mismas. Tras la evaluación del proceso de corrección de errores se detectó que la metodología utilizada obligaba a invertir mucho tiempo en la corrección de los mismos. Para la corrección de los subtítulos se utilizaba la herramienta Aegisub[28], un programa de edición de subtítulos gratuito que permite modificar tanto el texto como los tiempos de sincronismo de los mismos. Sin embargo, uno de los principales problemas que presenta este programa es que tanto su interfaz como su flujo de trabajo no permiten aprovechar la información adicional a la transcripción que se obtiene de un RAH y que sirve de ayuda en la corrección. Herramientas como Aegisub no están implementadas para ofrecer alternativas de corrección, entre otras soluciones, que puedan ser útiles en la corrección en diferido de las transcripciones; o, por otro lado, mientras que las aplicaciones de subtitulado automático funcionan tanto en diferido como en tiempo real, las herramientas de corrección de texto de propósito general, no están preparadas para recibir un flujo de datos (subtítulos) continuo y en tiempo real para su corrección. Como se verá en el Capítulo 6 de Pruebas y Resultados, la utilización de una aplicación de edición de textos que esté especialmente diseñada para la corrección de transcripciones generadas por un sistema de RAH reduce el tiempo empleado en la corrección de las mismas.

Dada la obligación de corregir la transcripción para ofrecer un subtitulado de calidad y para poder reentrenar el sistema mediante *enrollment*, la segunda parte del proyecto se centra en el estudio del API del Dragon NaturallySpeaking para optimizar el proceso de reconocimiento de voz y la posterior edición de la transcripción obtenida

[28]www.aegisub.org

como resultado de este proceso. Con tal fin, se modifica el programa APEINTADragon incorporando información adicional, como alternativas a nivel de frase, hipótesis de palabra, índices de confianza, etc., que responden a las necesidades planteadas para asistir al proceso de edición de la transcripción. Además y fruto de esta investigación, se han añadido otras funcionalidades que pretenden minimizar otros problemas relacionados con el RAH.

Con el fin de utilizar la nueva información complementaria al texto transcrito se ha optado por almacenarla en un archivo XML, de modo que pueda ser accedida con facilidad y con independencia de la herramienta que se utilice posteriormente.

En el siguiente capítulo se realiza una explicación más detallada de todas las prácticas realizadas, un análisis de los resultados y se plantean las conclusiones, así como su justificación, con el fin de alcanzar los dos objetivos planteados en este proyecto.

5. DESCRIPCIÓN DE LOS TRABAJOS REALIZADOS

En este capítulo se describirán los trabajos realizados para alcanzar los objetivos de este proyecto. Como se indicó en el capítulo anterior, el proyecto está dividido en dos partes, ambas basadas en el proyecto APEINTA:

- Estudio de la influencia del reentrenamiento en el subtitulado automático para reducir el WER.
- Optimización del proceso de corrección de errores.

5.1. EVALUACIÓN DEL WER EN EL SUBTITULADO AUTOMÁTICO DE VÍDEOS GENERADO POR UN SISTEMA DE RAH CON REENTRENAMIENTO

El objetivo de esta primera parte es analizar cómo el número de errores (Word Error Rate, WER) que se cometen en el proceso de reconocimiento de la voz se ven reducidos en sistemas de RAH que pueden ser reentrenados.

Como ya se introdujo en capítulos anteriores, los errores cometidos por el reconocedor de voz al realizar la identificación de los fonemas de la señal de voz tiene como consecuencia la escritura de fonogramas que no coinciden con lo expresado oralmente. Uno de los principales problemas de estos errores es que dificultan la comprensión del mensaje y reduce la utilidad del sistema. Por esta razón, funcionalidades como el reentrenamiento pueden ayudar a reducir estos errores cometidos y así mejorar la calidad del reconocimiento.

5.1.1. REENTRENAMIENTO

El reentrenamiento de un sistema de RAH adapta el perfil de voz de un locutor específico –perfil genérico o que el usuario ha creado a partir de un entrenamiento previo-, de modo que, además de introducir nuevo vocabulario, los sistemas acústico y de lenguaje se perfeccionan de acuerdo a sus características fónicas. Una mejora de estos sistemas repercute en una mayor precisión en el reconocimiento y por ende, una reducción del número de errores. En este caso, el reentrenamiento realizado se denomina en inglés *enrollment* y no precisa la presencia del locutor o usuario del perfil, a diferencia del reentrenamiento directo –en el que el mismo locutor es el que debe leer los textos de forma presencial para reentrenar el perfil. Éste se lleva a cabo utilizando un archivo de audio que contenga la voz del locutor y la transcripción literal de la misma y puede ser realizado tantas veces como material audiovisual nuevo se tenga.

Para mejorar y optimizar el proceso de enrollment se ha optado por sustituir la transcripción literal que se utiliza junto con el archivo de audio por una versión corregida de sí misma, es decir, libre de errores de transcripción.

La obligación de obtener un fichero corregido para mejorar la eficiencia del reentrenamiento supone la incorporación de una persona editora que realice las correcciones de cada una de las transcripciones. Esta fase se observó que consumía mucho tiempo ya que las aplicaciones utilizadas no están optimizadas para corregir grandes archivos de subtítulos o texto plano y no incluyen ninguna herramienta para asistir en esta tarea. Para el caso de la corrección de los subtítulos se utilizó Aegisub, una herramienta que permite la edición de los mismos. Como consecuencia, el tiempo medio necesario para corregir 15 minutos de vídeo se situó en una hora. La alternativa a la corrección de los subtítulos generados automáticamente es la generación de éstos desde cero. Sin embargo esta alternativa tiene una serie de desventajas sobre la primera: en pruebas anteriores, este sistema había reportado tiempos más altos aun, y para subtitular una 5 minutos de vídeo es necesario aproximadamente 1 hora de trabajo, lo que implica tres veces más de tiempo; por otro lado, la elección de esta alternativa respecto de la anterior obliga a invertir recursos en la formación de personas para cualificarlas para esta tarea. Ante estos datos, la corrección de los subtítulos obtenidos del reconocimiento automático es más eficiente en tiempo, requiere invertir menos recursos en formación y se convierte en la única alternativa para preparar las transcripciones para el reentrenamiento.

Para las pruebas de reentrenamiento y análisis del WER se ha utilizado el material audiovisual obtenido de las asignaturas de Catalogación de Materiales Especiales (CME) y Automatización de Centros y Servicios de Información (ACSI) de la diplomatura de Biblioteconomía y Documentación de la Universidad Carlos III de Madrid (Fig. 13). Cinco vídeos corresponden a la asignatura de ACSI, impartida por un varón (*Profesor 1*) y siete vídeos a la asignatura de CME, donde la locutora es una mujer (*Profesor 2*). La elección de dos asignaturas con profesorado de distinto sexo se ha hecho con la intención de evaluar si el reentrenamiento del sistema de reconocimiento de voz tiene los mismos resultados en ambos casos. Dado que la frecuencia fundamental de la voz femenina y masculina es distinta y la literatura existente indica que frecuencias fundamentales distintas pueden modificar la precisión del reconocedor [20], se pretende observar en los resultados del siguiente apartado si existe alguna variación de los valores del WER de cada Profesor.

Figura 13.Evaluación de APEINTA en la diplomatura de Biblioteconomía y Documentación de la UC3M.

Para crear los perfiles de los *Profesores 1* y *2* se emplearon 3 y 2 horas de entrenamiento inicial. En 2/3 del tiempo invertido el profesor realizó la lectura de textos con vocabulario común y el resto del tiempo se invirtió en la lectura de vocabulario técnico propio de la asignatura.

Como se puede ver en la Tabla 1, tras el entrenamiento inicial, los perfiles de ambos profesores fueron sometidos en tres ocasiones a un reentrenamiento por enrollment, dando lugar a tres nuevas versiones del perfil de cada profesor. El reentrenamiento se realizó con los vídeos grabados en las clases de las asignaturas de cada uno de los profesores.

El perfil V1 se obtuvo de reentrenar el perfil V0 con nuevo material audiovisual; el perfil V2 se generó de la misma forma pero a partir del perfil V1; por último, el perfil V3 se creó a partir del V2. Además de las versiones producto del reentrenamiento (V1, V2 y V3), cada profesor tiene una versión del perfil sin entrenar (VNT), más otra que cuenta con solamente con el entrenamiento inicial (V0).

Enrollment \ Versión del Perfil	VNT	V0	V1	V2	V3
Ningún entrenamiento	X	X	X	X	X
Entrenamiento básico		X	X	X	X
Enrollment 1			X	X	X
Enrollment 2				X	X
Enrollment 3					X

Tabla 1. Enrollment del Profesor 1 y Profesor 2.

Para realizar el reentrenamiento de DNS se debe indicar por consola el nombre del fichero que contiene la ruta del texto y el audio (Fig. 14) que se utilizarán para reentrenar y el perfil de voz del usuario que se desea actualizar:

efenroll.exe -username="nombre-perfil" -basetopic="Español (España) | BestMatch Plus | General" -inputfile="profesor2.txt" -baseAM="Español (España) | BestMatch III"

Donde:

-username: indica el nombre del perfil que será reentrenado.

-basetopic: idioma del vocabulario, modelo del vocabulario y tipo de diccionario del idioma seleccionado.

-inputfile: archivo de configuración que contiene la ruta del fichero de audio y del texto transcrito corregido.

Ejemplo de fichero .txt de configuración (Fig. 14):

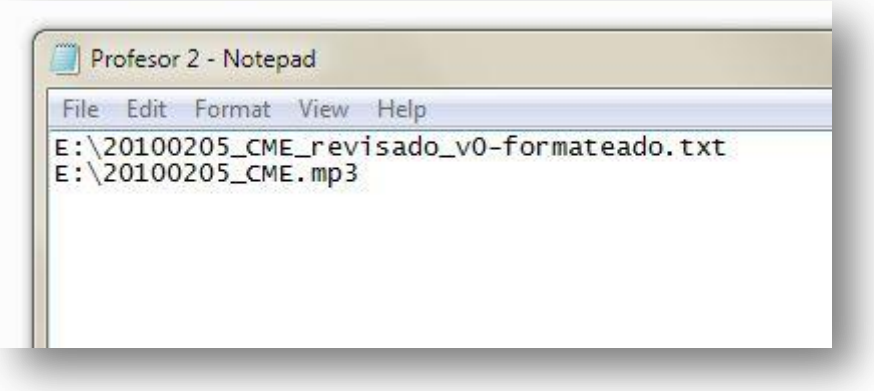


Figura 14. Fichero de configuración para enrollment con la ruta de los archivos a utilizar.

-baseAM: idioma del vocabulario y tipo de modelo acústico a utilizar.

Para obtener las transcripciones que corresponden a cada uno de los vídeos se ha usado APEINTAdragon, utilizando la función de transcripción desde fichero y aplicando, en cada caso, el perfil correspondiente a cada profesor.

APEINTAdragon (Fig. 15) es un programa desarrollado dentro del marco del proyecto APEINTA. Este programa es donde se lleva a cabo el proceso de reconocimiento de voz y permite la generación automática de recursos educativos –como subtítulos, texto plano, etc.- a partir de un fichero de audio o de la entrada de un micrófono.

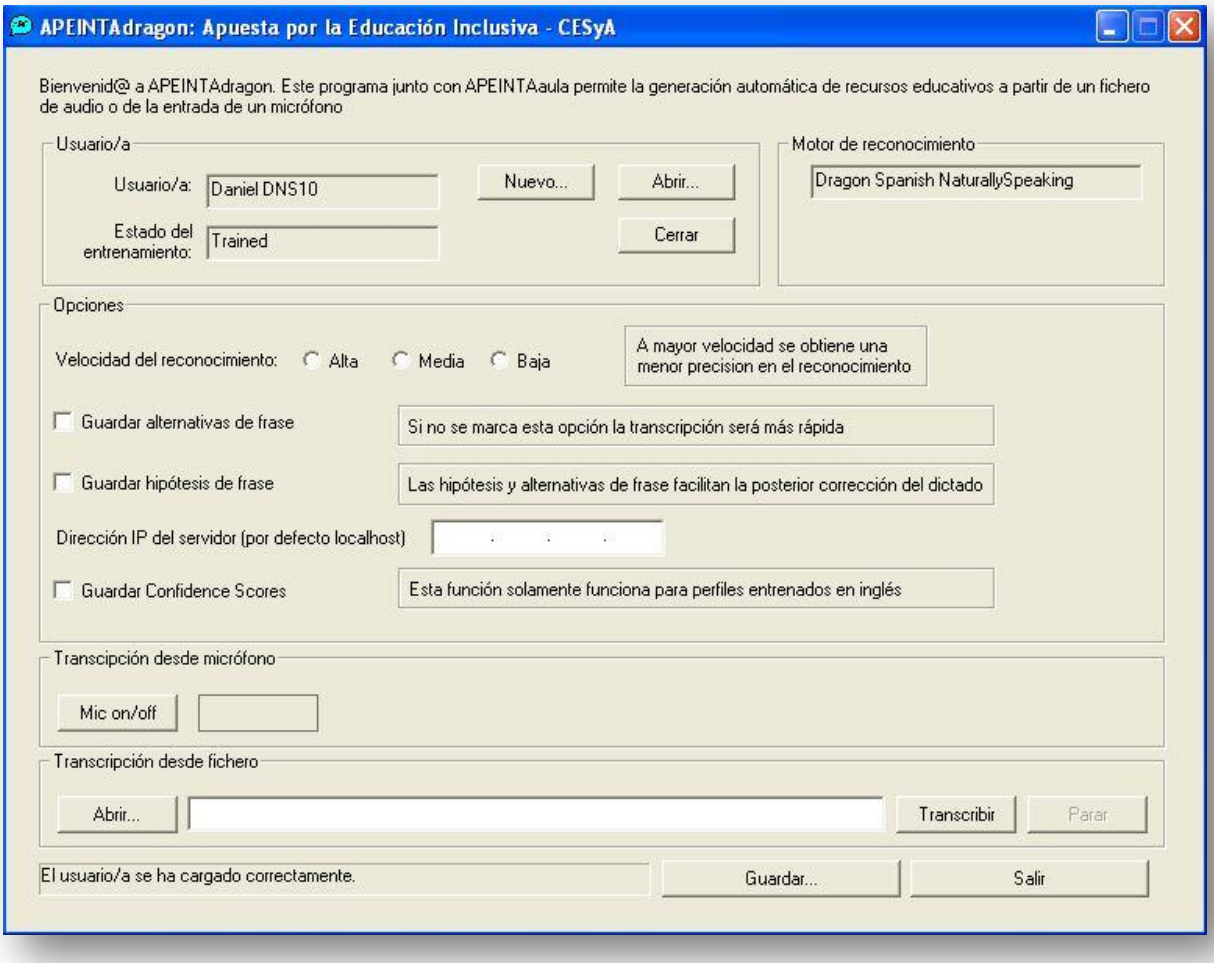


Figura 15. Interfaz de APEINTADragon.

Tras realizar el reconocimiento de cada uno de los 12 vídeos –cinco vídeos del *Profesor 1* y siete del *Profesor 2*-, se han obtenido un total de 60 transcripciones –teniendo en cuenta los cinco perfiles que tienen *Profesor 1* y *Profesor 2*. Estas transcripciones se han obtenido en forma de texto plano y con una codificación UTF-8 para su posterior análisis.

5.1.2. ANÁLISIS DE LA EVOLUCIÓN DEL WER

El WER (Word Error Rate) es una medida utilizada en los sistemas de reconocimiento del habla que calcula el número de errores como la suma de inserciones, borrados y sustituciones de una palabra por otra.

Para calcular el WER de cada una de las transcripciones se ha utilizado la herramienta SCLITE del programa Speech Recognition Scoring Toolkit (SCTK) Version 2.4.0 [4] del National Institute of Standards and Technology (NIST). Este programa evalúa la eficiencia de un sistema de RAH mediante la comparación de la transcripción

resultante de un proceso de reconocimiento de voz –texto hipótesis- con una transcripción idéntica pero corregida –texto referencia.

El SCLITE utiliza un algoritmo basado en la Distancia de Levenshtein, que mide el número de operaciones –sustitución, borrado o inserción- necesarias para transformar una cadena de caracteres en otra distinta, para relacionar las palabras del texto hipótesis con las del de referencia (Dynamic Programming Algorithm) [21]. La métrica elegida no tiene en cuenta los pesos de las palabras, es decir, la importancia o influencia que estas pueden tener dentro del texto y el efecto que tiene su modificación sobre la comprensión del mensaje transmitido. Esto se debe a que se pretende evaluar el sistema en su conjunto dándole la misma importancia a todas las palabras presentes en las transcripciones. De este modo, las inserciones, borrados y sustituciones tienen el mismo valor en la función que calcula el WER y su cálculo se realiza según la fórmula:

$$WER = \frac{S + B + I}{N}$$

Donde

- S es el número de sustituciones
- B es el total de palabras borradas
- I es el sumatorio de inserciones
- N es el total de palabras que tiene el texto de referencia

Para obtener los valores de cada uno de los tipos de error de las 60 transcripciones se ejecuta la herramienta SCLITE con la siguiente sentencia en consola:

sclite –r {archivo-de-referencia} –h {archivo-hipótesis} {opciones}

Donde el archivo de referencia es el mismo para cada vídeo y el archivo hipótesis es aquel resultante del reconocimiento bajo cada uno de los cinco perfiles de cada profesor.

El informe de resultados que provee SCLITE se muestra en las Fig. 16 y 17. En la primera figura, Fig. 16, se han calculado los errores y se han clasificado atendiendo a su tipología. De este modo, se muestra el número de errores de sustitución (*Sub*), borrado (*Del*) e inserción (*Ins*) y el sumatorio de todos ellos (*Err*), para cada una de las transcripciones resultantes de haber utilizado los cinco perfiles sobre un mismo vídeo. El número de palabras de la transcripción aparece bajo la columna *Ref*, mientras que el número de palabras que contiene cada una de las transcripciones sin corregir están en la columna *Corr* de la Fig. 16. Las transcripciones, de los distintos perfiles, se listan en la columna *System*.

En la Fig. 17 aparecen las mismas medidas que las explicadas en la Fig. 16, sin embargo, en ésta los valores se encuentran en porcentaje en lugar de valor absoluto. Además, en la Fig. 17 se incluye un ejemplo del resultado del cálculo de la media (*mean*),

desviación típica (*std dev*) y el valor máximo y mínimo del WER alcanzado para un vídeo determinado.

Executive Scoring Summary by Word Tokens

System	# Snt	# Ref	Corr	Sub	Del	Ins	Err
./carlos/20100326_ASCI/20100326_NT/20100326_NT.txt	1	5884	4660	770	454	238	1462
./carlos/20100326_ASCI/20100326_v0/20100326_v0.txt	1	5884	4857	601	426	232	1259
./carlos/20100326_ASCI/20100326_v1/20100326_v1.txt	1	5884	5183	350	351	165	866
./carlos/20100326_ASCI/20100326_v2/20100326_v2.txt	1	5884	5084	505	295	205	1005
./carlos/20100326_ASCI/20100326_v3/20100326_v3.txt	1	5884	5250	350	284	163	797

Figura 16. Informe de resultados del SCLITE.

Executive Scoring Summary by Percentages

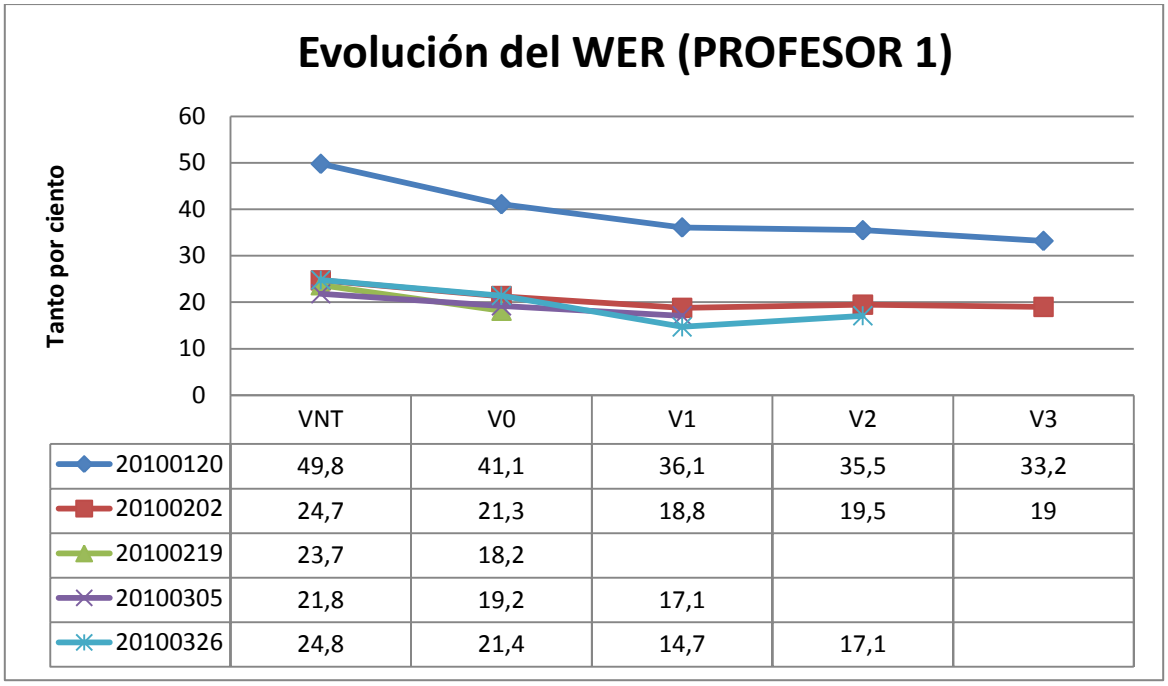
System	# Snt	# Ref	Corr	Sub	Del	Ins	Err
./carlos/20100326_ASCI/20100326_NT/20100326_NT.txt	1	5884	79.2	13.1	7.7	4.0	24.8
./carlos/20100326_ASCI/20100326_v0/20100326_v0.txt	1	5884	82.5	10.2	7.2	3.9	21.4
./carlos/20100326_ASCI/20100326_v1/20100326_v1.txt	1	5884	88.1	5.9	6.0	2.8	14.7
./carlos/20100326_ASCI/20100326_v2/20100326_v2.txt	1	5884	86.4	8.6	5.0	3.5	17.1
./carlos/20100326_ASCI/20100326_v3/20100326_v3.txt	1	5884	89.2	5.9	4.8	2.8	13.5

SPKR	high	low	std dev	mean
	24.8	13.5	4.7	18.3

Figura 17. Informe de resultados del SCLITE.

A partir de estos resultados se han realizado las gráficas de las Fig. 18, 19 y 20, en las que se analiza la evolución del WER de los diferentes vídeos a medida que se realiza un nuevo reentrenamiento. En la Fig. 21 se muestra cómo varía la tipología del error cuando un perfil ha sido reentrenado y mejorado.

En el caso de *Profesor 1* los resultados del WER obtenidos a partir de los cinco perfiles aplicados a los cinco vídeos se ven en la Fig. 18:



	VNT	V0	V1	V2	V3
20100120	49,8	41,1	36,1	35,5	33,2
20100202	24,7	21,3	18,8	19,5	19
20100219	23,7	18,2			
20100305	21,8	19,2	17,1		
20100326	24,8	21,4	14,7	17,1	

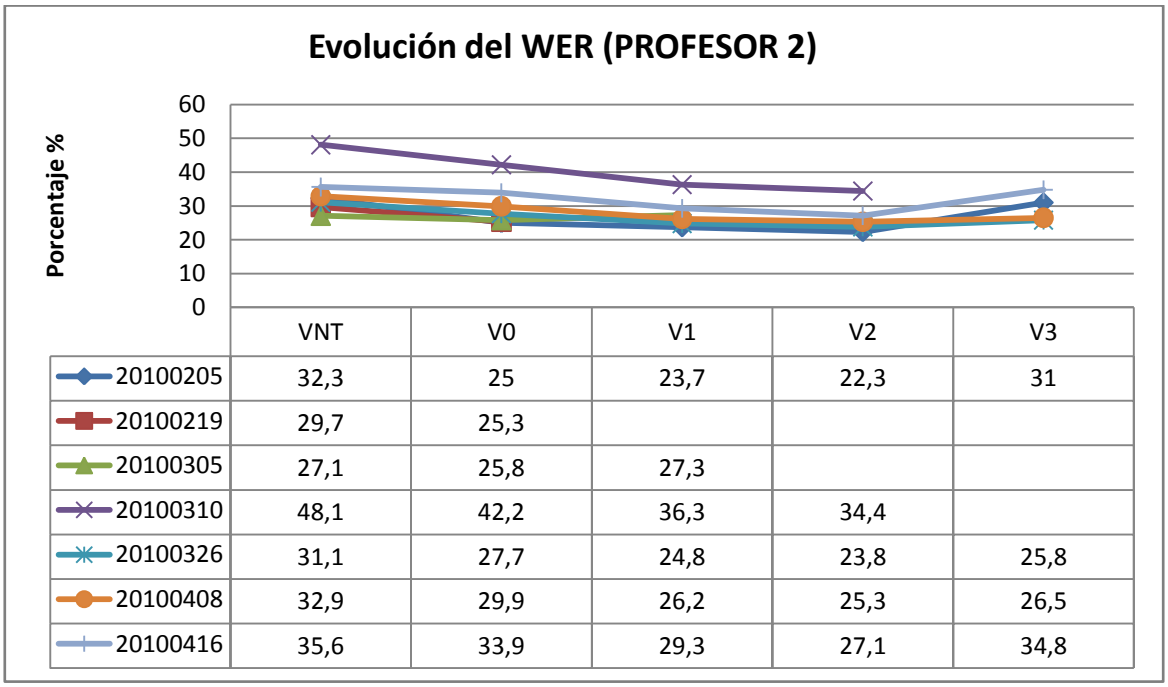
Figura 18. Evolución del WER.

Cada color mostrado en la Fig. 18 se corresponde con un vídeo determinado, así como el nombre de cada vídeo responde a la fecha en el que este fue grabado. Cada vídeo tiene hasta cinco valores distintos que corresponden al WER obtenido de haber sometido a ese vídeo a un proceso de reconocimiento utilizando un perfil determinado. Los cinco perfiles son distintos entre sí. VNT es un perfil que no ha tenido entrenamiento alguno; V0, se corresponde con un perfil que tiene un entrenamiento inicial básico –sin reentrenamiento-; el resto de los perfiles, V1, V2 y V3 que además del entrenamiento inicial han sido reentrenados y por lo tanto adaptados a las características de cada profesor.

Los vídeos que han sido utilizados para realizar *enrollment* a un perfil determinado no han sido utilizados para la fase de test posterior. De este modo el vídeo 20100219, que se utilizó para hacer el *enrollment* del V0 y generar el perfil V1 no se ha usado como material de prueba para calcular su WER, dado que podría falsear los resultados. Debido a que los perfiles V2 y V3 han sido obtenidos a partir de V1, tampoco se utilizará el vídeo 20100219 para la prueba de estor perfiles. Ocurre lo mismo con los vídeos 20100305 y 20100326, estos se han utilizado para generar los perfiles V2 y V3 respectivamente, por lo que no se han usado para calcular su WER en esos perfiles. En definitiva, las casillas en blanco corresponden al WER de los vídeos que han sido utilizados para realizar el *enrollment* de perfiles, de modo que no se pueden usar como material para evaluar el sistema. La elección de estos tres vídeos para reentrenar no tiene una razón técnica, ya que tienen una longitud similar, en torno a los 45 minutos. Si no que responde más bien a unos criterios de disponibilidad para poder proseguir con el curso de la investigación.

Corroborando las tesis expuestas sobre la importancia del reentrenamiento para reducir la tasa de error, se puede observar como la tendencia del WER es verse reducida a medida que el perfil es reentrenado. La reducción más notable se encuentra en la transición de V0 a V1, la cual corresponde al primer enrollment realizado. Los reentrenamientos sucesivos reducen, aunque en menor medida, los niveles de error y como consecuencia permiten obtener un mínimo global del error en V3. También es importante indicar cómo la mayor reducción del error se produce cuando el perfil del *Profesor 1* pasa VNT –perfil sin ningún tipo de entrenamiento- a V0 –con un entrenamiento inicial. Esto demuestra cómo de importante resulta el entrenamiento inicial de los perfiles. Por otro lado, el vídeo 20100120 se observa que tiene unos valores más altos que el resto. Esto se debe a que puntualmente en ese vídeo se grabó mucho ruido en la señal de audio – mala relación señal a ruido (SNR)-, observando que el reconocedor del habla utilizado es muy sensible en estos casos y produce peores valores de WER. Aunque se verá más en detalle con el *Profesor 2*, este resultado demuestra la importancia de que la señal acústica tenga una buena SNR para conseguir mejores resultados en el reconocimiento.

En el caso del *Profesor 2*, los resultados arrojados tras aplicar el SCLITE a cada una de las transcripciones obtenidas de cada uno de los siete vídeos son los siguientes (Fig. 19):



	VNT	V0	V1	V2	V3
20100205	32,3	25	23,7	22,3	31
20100219	29,7	25,3			
20100305	27,1	25,8	27,3		
20100310	48,1	42,2	36,3	34,4	
20100326	31,1	27,7	24,8	23,8	25,8
20100408	32,9	29,9	26,2	25,3	26,5
20100416	35,6	33,9	29,3	27,1	34,8

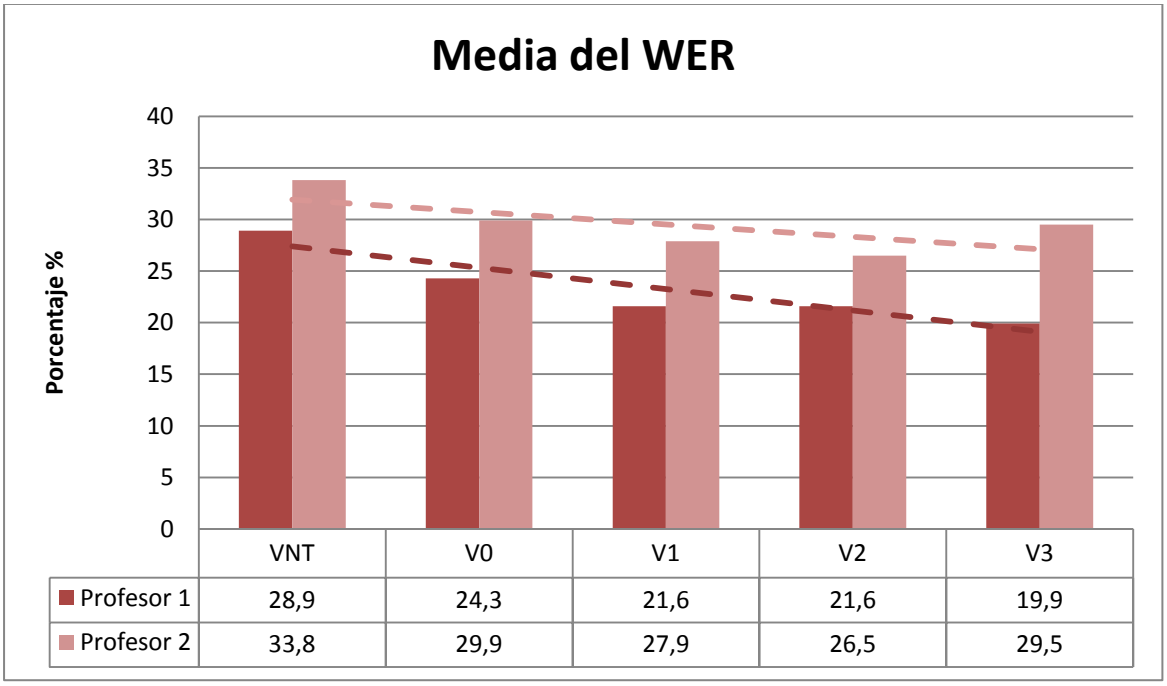
Figura 19. Evolución del WER.

Al igual que en el caso anterior, los vídeos utilizados para crear nuevos perfiles a partir del anterior por enrollment no han sido utilizados para evaluar la eficiencia del sistema de RAH. En este caso, el vídeo 20100219 se utilizó para obtener el perfil V1, el vídeo 20100305 para reentrenar el perfil V1 y obtener V2 y el vídeo 20100310 para calcular el perfil V3 a partir de V2.

En este caso, también se observa como a medida que se realizan nuevos reentrenamientos del perfil la tasa de error disminuye. Esto es cierto menos para el perfil V3, en el que el WER no disminuye sino que se estabiliza o aumenta. En este último caso el *enrollment* realizado sobre V2 para generar V3 se hizo eligiendo un vídeo (vídeo 20100310 en color violeta) en el cual la calidad –desde el punto de vista de los requisitos para obtener un buen reconocimiento- de la clase impartida por el *Profesor 2* fue bastante deficiente. En este caso confluyeron tres factores críticos que determinan la calidad del reconocimiento. En primer lugar el *Profesor 2* se situó muy lejos del micrófono por lo que se obtuvo una mala calidad del audio (baja SNR); en segundo lugar, la velocidad en el discurso fue muy elevada, lo que dificulta la identificación de los sonidos; y por último, el discurso contenía frases muy cortas y carecían de un hilo narrativo -la forma de hablar influye en la calidad del reconocimiento [19]. Este último aspecto es el más crítico de todos y el causante de valores tan altos del WER; se puede ver como los valores del WER, para este vídeo, en los perfiles VNT, V0, V1 y V2 son notablemente superiores al resto de valores del WER para esos perfiles en los demás vídeos.

Como consecuencia de utilizar el vídeo 20100310 para reentrenar V2, el perfil resultante V3 no aporta ninguna mejora sino que provoca peores tasas de error para aquellos vídeos transcritos con ese perfil, de ahí el repunte de la tasa de error en todas las gráficas. Se puede concluir entonces, que si el material utilizado para reentrenar tiene una calidad y parámetros similares y de buena calidad los perfiles y el reconocimiento mejorarán; de lo contrario, podrán empeorar.

Mezclando los datos de ambos profesores, en la siguiente figura se observa la evolución de la media del WER de cada profesor según el perfil utilizado (Fig. 20).



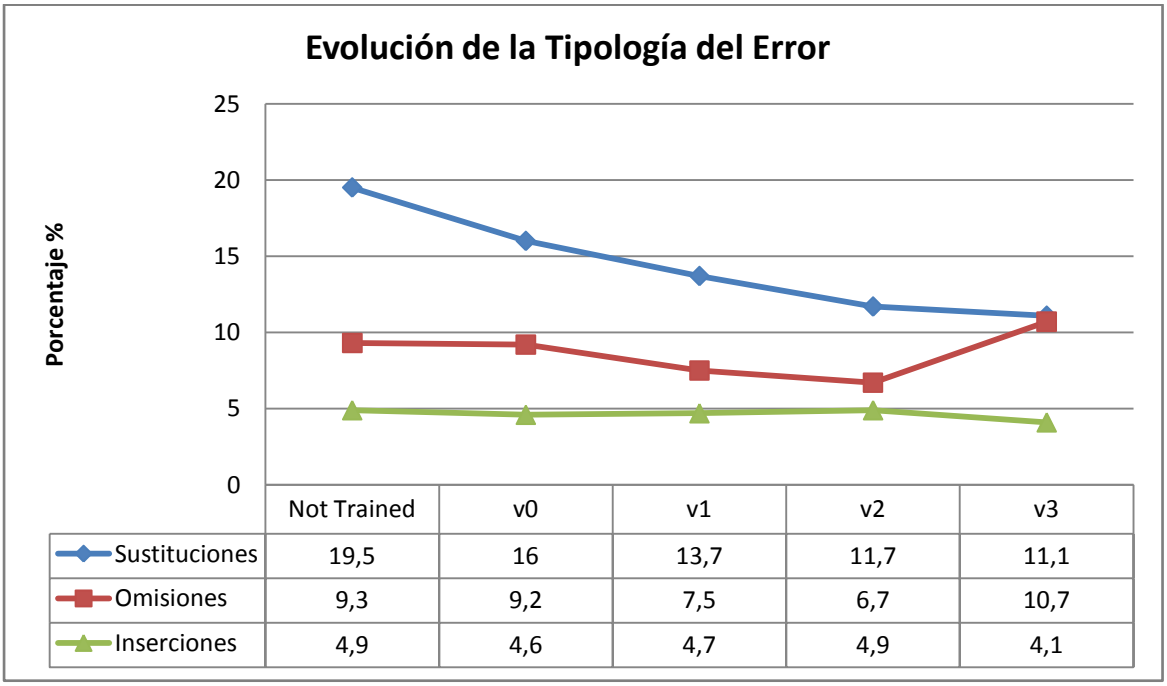
	VNT	V0	V1	V2	V3
Profesor 1	28,9	24,3	21,6	21,6	19,9
Profesor 2	33,8	29,9	27,9	26,5	29,5

Figura 20. Evolución de la media del WER de cada profesor según el perfil utilizado.

En este caso, se puede ver como el valor del WER de ambos profesores se reduce a medida que el perfil es reentrenado y, por tanto, V3 tiene una tasa de error más baja que V0 –primera versión del perfil con entrenamiento inicial. Aunque la tasa de error del *Profesor 2* es sensiblemente superior a la del *Profesor 1* se puede ver como la reducción del WER en ambos casos entre el perfil VNT y V3 es similar, 9 y 7,3 -si utilizamos el perfil V2 del *Profesor 2*, ya que el V3 es el perfil defectuoso- puntos respectivamente.

Al igual que lo indicado en la literatura, la diferencia de la frecuencia fundamental entre ambos profesores, también puede haber sido un factor a tener en cuenta a la hora de variar la precisión del reconocedor, modificando el WER. Sin embargo, dada el pequeño volumen de vídeos para realizar las pruebas no se puede concluir que en tono sea un factor fundamental en este caso.

Si analizamos la evolución media de cada tipo de error en ambos profesores, en la Fig. 21 se puede ver cómo se reduce la tasa de cada uno de los tipos de error a medida que se reentrena el sistema de voz. La evolución de los errores según su tipología es muy similar en ambos Profesores por lo que se ha optado por hace una media y mostrarla en una sola gráfica.



	Not Trained	v0	v1	v2	v3
Sustituciones	19,5	16	13,7	11,7	11,1
Omisiones	9,3	9,2	7,5	6,7	10,7
Inserciones	4,9	4,6	4,7	4,9	4,1

Figura 21. Evolución del tipo de error según el perfil utilizado.

En la Fig. 21 se observa cómo las sustituciones y las omisiones se reducen a medida que mejora el perfil utilizado, mientras que las inserciones, un error con una incidencia mucho menor, se mantienen constantes en los cinco perfiles. Destacar como el perfil V3, el perfil "problemático" del *Profesor 2*, provoca un aumento de la tasa de omisiones, mientras que el resto de errores no se ve alterado. De los resultados

ofrecidos por este gráfico se entiende que el tipo de error más sensible a perfiles que provocan peores WER, como es el caso del V3 del *Profesor 2*, es el de las omisiones.

5.1.3. CONCLUSIONES

Según los resultados obtenidos se puede observar que el reentrenamiento, en este caso por enrollment, ayuda a reducir la tasa de errores cometidos durante el reconocimiento, reduciendo principalmente la tasa de sustituciones y omisiones. Si bien el reentrenamiento mejora el WER, la elección del material audiovisual para realizarlo condiciona notablemente la calidad del nuevo perfil generado. Como se ha visto en este capítulo, el reentrenamiento de un perfil a partir de un vídeo con unas características de sonido pobres –baja SNR- y donde su discurso no cumple los requisitos recomendados para un buen reconocimiento –baja velocidad del habla y construcción de frases largas y completas-, provocan que el nuevo perfil tenga una calidad inferior al anterior y que los vídeos transcritos a partir de éste den una tasa de WER mayor que el obtenido por el resto de perfiles con menor reentrenamiento.

Por otro lado, el reentrenamiento prueba que la utilización de sistemas de subtitulado automático como APEINTA es más efectiva que realizar el subtitulado del material audiovisual desde cero, tanto por el tiempo como por los recursos invertidos. Siendo el tiempo de corrección/creación un aspecto a tener en cuenta, en el segundo caso no existe posibilidad de reducir drásticamente el tiempo invertido, mientras que en el caso de la generación automática de subtítulos sí puede verse reducido. A medida que el reentrenamiento mejora la precisión del sistema reconocedor, éste introduce menos errores y, por lo tanto, el tiempo de corrección disminuye progresivamente.

Sin embargo, se ha detectado que la corrección de los errores de las transcripciones generadas por los sistemas de subtitulado automático es un aspecto crítico en el proceso global y la metodología utilizada es subóptima. Es por esto que la segunda parte del proyecto se centra en encontrar soluciones que asistan en esta tarea y, para ello, se tratará de modificar el sistema reconocedor para obtener información adicional que apoye y facilite la corrección de las transcripciones.

5.2. MODIFICACIÓN DEL SISTEMA DE RAH PARA OPTIMIZAR EL RECONOCIMIENTO DE LA VOZ Y FACILITAR LA CORRECCIÓN DE ERRORES

Para obtener nueva información que asista en el proceso de corrección de las transcripciones se ha estudiado el API de DNS. En este capítulo se describen las nuevas funcionalidades relacionadas con el proceso de edición que han sido incorporadas al programa APEINTADragon. Fruto de esta investigación se han incorporado otras funciones útiles de cara a otros problemas detectados en el conjunto del proceso.

5.2.1. ESTUDIO DEL API DE DRAGON

En una primera versión, APEINTAserver obtenía únicamente el texto reconocido, los tiempos de inicio y fin de cada frase y el número de alternativas a la frase de referencia (Fig. 22).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CLASE>
    <PROFE>
        <TEXT>erradicar el hambre en el mundo es uno de los objetivos del milenio de la ONU</TEXT>
        <TIME_IN>1295521490385</TIME_IN>
        <TIME_OUT>1295521494825</TIME_OUT>
        <NUM_ALT>30</NUM_ALT>
    </PROFE>
</CLASE>
```

Figura 22. Documento *.XML* generado por APEINTAserver.

Esta información es suficiente y válida para cubrir las necesidades de APEINTA, subtitulado automático y sincronizado en dos escenarios propuestos: subtitulado en directo y en diferido. Sin embargo, para dar solución a las necesidades planteadas en el capítulo anterior, mejorar el proceso de corrección de errores en un escenario de subtitulado en diferido, la información relativa al reconocimiento que hasta el momento ofrece APEINTADragon es escasa y requiere obtener nueva información que pueda ser de utilidad en el proceso de corrección y permita reducir el tiempo invertido en esta fase.

Para ello se decide estudiar el API que ofrece Dragon bajo la licencia de Dragon Software Development Kit Client Edition en su versión 9.5, compatible con Windows XP. Ésta ofrece librerías tanto en Microsoft C++ como en VisualBasic. Para la programación de las nuevas funciones se ha utilizado Microsoft C++, dado que este proyecto utiliza la aplicación APEINTADragon para su optimización y ésta está programada es este lenguaje. Como complemento al API de Dragon se ha programado usando el API Win32, marco de trabajo basado en C.

Para la programación de las aplicaciones de este proyecto se ha utilizado el IDE, entorno de desarrollo integrado, Visual Studio 2005 en su versión 8.0, cuya integración con el lenguaje de programación utilizado era muy buena.

5.2.1.1. DIAGRAMA DE CASOS DE USO

En la Fig. 23 se muestra el diagrama de casos de uso implementado en APEINTADragon donde el usuario, donde el locutor utiliza el sistema de subtitulado para obtener la transcripción.

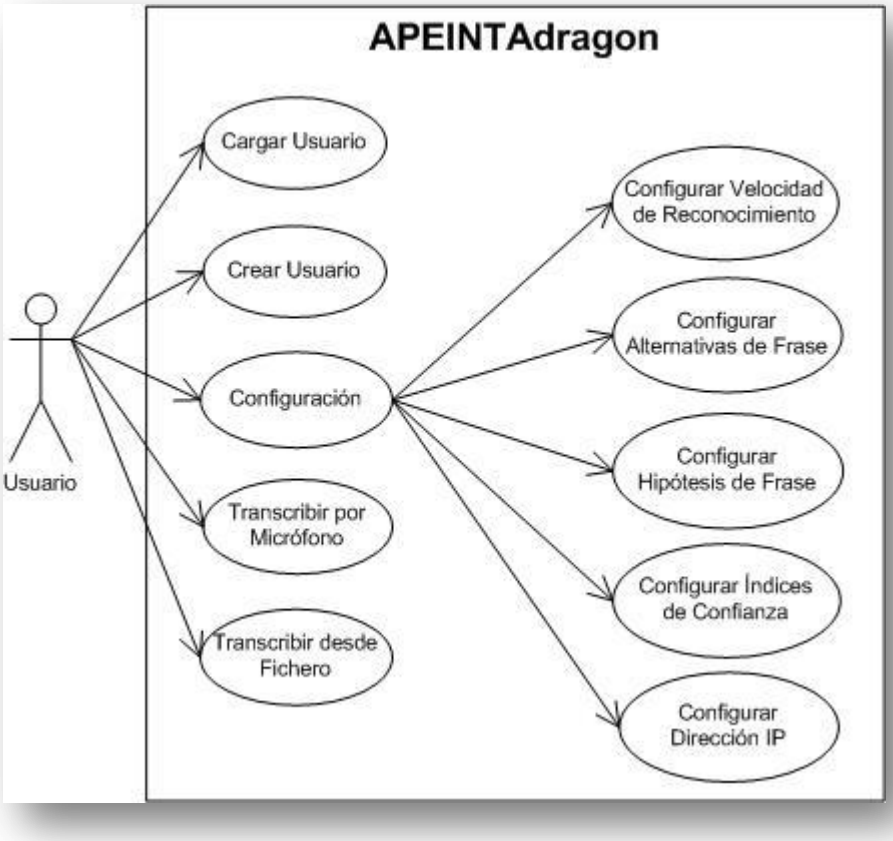


Figura 23. Diagrama de casos de usos de la aplicación APEINTAdragon.

En la Fig. 23 se muestran los distintos casos de uso que presenta el sistema. El sistema comienza con la elección de un perfil que haya sido previamente entrenado (Cargar Usuario) o bien, opta por crear un nuevo perfil si el usuario no tuviera ninguno propio (Crear Usuario). Una vez hecho esto, se procede a configurar la aplicación APEINTAdragon mediante la elección de las diferentes opciones que se ofrecen (Configurar Velocidad de Reconocimiento, Configurar Alternativas de frase, Configurar Hipótesis de frase, etc.). Finalmente, el usuario puede elegir entre transcribir un fichero de audio previamente grabado, o bien, mediante la entrada de un micrófono transcribir la señal voz generada en directo.

5.2.1.2. FUNCIONES INCORPORADAS A LA APLICACIÓN APEINTADRAGON

Las nuevas funcionalidades con las que cuenta la nueva aplicación son:

- Obtención de alternativas a la frase de referencia.
- Cómputo de los tiempos de inicio y fin por palabra.
- Cálculo del índice de confianza de cada palabra.
- Grabación del audio capturado como ayuda para la corrección.
- Reducción del tiempo invertido en el proceso de reconocimiento (retardo).
- Obtención de hipótesis de palabra previas a que finalice la transcripción.
- Filtrado de sonidos cuya transcripción dificultan la comprensión del texto.
- Incluir signos de puntuación en el texto resultante de la transcripción.
- Generación de archivos .XML y .DRA con la información obtenida.

Frases alternativas

Una de las principales ayudas a la corrección de subtítulos pasa por ofrecer alternativas a las frases reconocidas (frases de referencia) y que se muestran en la transcripción.

El reconocedor incluye la información de la que ha hecho uso para construir la frase de referencia y la utiliza para construir las frases alternativas. La obtención de las frases alternativas –entre 0 y 30 por cada frase de referencia- depende directamente de la calidad del discurso. De la misma forma que el WER aumentaba notablemente ante un tipo de discurso de frases muy cortas, con poca conexión y muy veloz, el número de frases alternativas también están condicionadas por estos factores.

La obtención de las frases alternativas se realiza al mismo tiempo y en el mismo evento que la frase de referencia –evento *makeChanges* de la clase *DgnDictCustom*- que se ejecuta cada vez que se detecta un silencio mayor a un segundo de duración; el texto que se encuentra en ese instante en el buffer es interpretado por el motor de reconocimiento como una frase.

La información relativa a la transcripción se almacena en un objeto *DgnLastResult –DgnLastResult* de la clase *DgnDictCustom*- al que se accederá en múltiples ocasiones para obtener datos como alternativas de palabra, tiempos de frase, scores, etc. En este caso, las frases alternativas junto con la de referencia se encuentran dentro del objeto *DgnResPhrases* de la clase *DgnDictCustom*.

Se ha incluido un botón en la interfaz de la aplicación para poder elegir si guardar o no las alternativas de frase (Fig. 24).

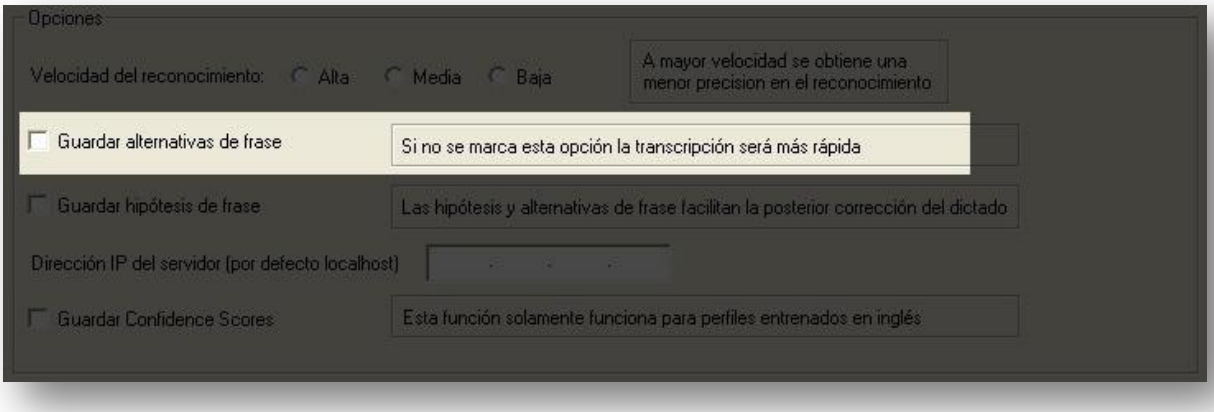


Figura 24. Cálculo de las alternativas de frase

El cálculo de las frases alternativas repercute directamente en el retardo. Aunque el retardo introducido es muy pequeño si se analiza a nivel de frase y no tiene por qué interferir en el subtitulado en directo, si no se requiere el uso de frases alternativas es recomendable evitar su cálculo, de forma que el motor de reconocimiento no tenga que calcular las frases y así reducir el tiempo invertido en el proceso. Si no se marca esta opción, a la constante *dgndictoptionNbestCount* de *DgnDictationOptionConstants* se le asigna el valor 1 lo que impide al motor de reconocimiento calcular las frases alternativas teniendo sólo en cuenta la primera opción. Esto es especialmente útil en situaciones de subtitulado en directo donde se necesita reducir el retardo al mínimo.

De lo contrario, en un escenario de subtitulado en diferido y donde el retardo no es un inconveniente, la obtención de frases alternativas puede servir de ayuda para que una aplicación de edición de transcripciones haga uso de éstas durante la corrección de la transcripción. En esta aplicación el usuario selecciona un conjunto de palabras erróneas del texto transcrito y si desea corregirlas puede presionar una tecla/ratón para que aparezca una lista con las palabras o frases alternativas y elegir una para sustituir las palabras erróneas del texto transcrito. Esta técnica, la cual puede ser objeto de estudio para su mejora en trabajos futuros, es más eficaz que el actual método utilizado puesto que requiere menos tiempo y acciones por cada corrección.

En la Fig. 25 se muestra un ejemplo de las frases alternativas a la frase de referencia “Antes en la vida renuncia”.

```
<RefWords numwrds="5">
  <WRD c="500" end="2340" id="1" srt="antes" start="1941"/>
  <WRD c="500" end="2460" id="2" srt="en" start="2340"/>
  <WRD c="500" end="2579" id="3" srt="la" start="2460"/>
  <WRD c="500" end="2998" id="4" srt="vida" start="2579"/>
  <WRD c="500" end="4355" id="5" srt="renuncia" start="3517"/>
</RefWords>
<AltPhrase>
  <ALT id="1" srt="la francesa nacida en diciembre"/>
  <ALT id="2" srt="antes en la vida denuncian"/>
  <ALT id="3" srt="la francesa nacida en un"/>
  <ALT id="4" srt="la concesionaria denuncian"/>
  <ALT id="5" srt="antes en la vida denuncia"/>
  <ALT id="6" srt="antes en la vida en un"/>
  <ALT id="7" srt="antes en la vida renuncian"/>
  <ALT id="8" srt="la concesionaria denuncia"/>
  <ALT id="9" srt="antes en la vida denuncian"/>
  <ALT id="10" srt="la concesionaria renuncia"/>
  <ALT id="11" srt="antes en la vida denuncia"/>
  <ALT id="12" srt="la concesionaria renuncia"/>
  <ALT id="13" srt="las concesionarias denuncian"/>
  <ALT id="14" srt="antes en la vida renuncia"/>
  <ALT id="15" srt="dije antes en la vida renuncia"/>
  <ALT id="16" srt="la concesionaria denuncian"/>
  <ALT id="17" srt="antes en la vida en diciembre"/>
  <ALT id="18" srt="dije antes en la vida denuncian"/>
  <ALT id="19" srt="antes de nacida en diciembre"/>
  <ALT id="20" srt="antes de Navidad renuncia"/>
  <ALT id="21" srt="la francesa nacida en un cien"/>
  <ALT id="22" srt="antes de nacida en un"/>
  <ALT id="23" srt="antes de Navidad denuncian"/>
  <ALT id="24" srt="la concesionaria denuncia"/>
  <ALT id="25" srt="antes en la vida en un"/>
  <ALT id="26" srt="la concesionaria renuncian"/>
  <ALT id="27" srt="la concesionaria enuncia"/>
  <ALT id="28" srt="la concesionaria en diciembre"/>
  <ALT id="29" srt="la francesa nacida en oposición"/>
</AltPhrase>
```

Figura 25. Frases alternativas

Tiempos de palabra

A través de los métodos *get_BeginningTime* y *get_EndingTime* del objeto *DngResWord* de la clase *DgnDictCustom* se obtienen los tiempos de inicio y fin de cada palabra. La información temporal de cada palabra es importante tanto para sincronizar el texto y el audio como, de cara al proceso de corrección, poder localizar y tener acceso a las palabras que se encuentran en un intervalo de tiempo determinado.

Índices de confianza

El índice de confianza –del término inglés confidence score- es el valor que asigna el reconocedor a cada palabra en función de la probabilidad de que ésta haya sido reconocida correctamente. Dragon asigna índices por valor de 0 hasta 1000 a las palabras de las frases de referencia, mientras que las palabras de las frases alternativas y las hipótesis de palabra no se puntúan con ningún valor.

En la Fig. 26 se puede ver como el atributo *c* contiene los índices de confianza de cada una de las palabras. Todas ellas tienen unos índices por encima de 980 –siendo 1000 el valor máximo- dado que su reconocimiento se considera acertado y, en efecto, corresponde a lo que se pronunció. Sin embargo, la octava palabra, *while*, tiene un índice de confianza de 788 que indica que existe incertidumbre en cuanto a su correcto reconocimiento; la palabra pronunciada fue *anual*.

```
<Phrase end="4801" id="1" start="13">
  <RefWords numwrds="9">
    <WRD c="992" end="1470" id="1" srt="good morning" start="871"/>
    <WRD c="996" end="1569" id="2" srt="and" start="1470"/>
    <WRD c="995" end="2048" id="3" srt="welcome" start="1569"/>
    <WRD c="997" end="2148" id="4" srt="to" start="2048"/>
    <WRD c="995" end="2268" id="5" srt="the" start="2148"/>
    <WRD c="992" end="2707" id="6" srt="first" start="2268"/>
    <WRD c="983" end="2926" id="7" srt="and" start="2727"/>
    <WRD c="788" end="3206" id="8" srt="while" start="2926"/>
    <WRD c="979" end="3685" id="9" srt="meeting" start="3206"/>
  </RefWords>
</Phrase>
```

Figura 26.Índice de confianza de las palabras de la frase "good morning welcome to the first while meeting".

Los índices se obtienen con el método *get_ConfidenceScore* en cada *DngResWord*.

En el ejemplo anterior se ha utilizado el inglés ya que durante la realización del proyecto se confirmó que Dragon solamente implementa esta funcionalidad para reconocimiento de voz con perfiles entrenados en inglés, asignando a las palabras transcritas con perfiles en el resto de idiomas un valor predeterminado de 500. Aunque los índices de confianza no se puedan utilizar en perfiles de idioma español, se ha incluido esta funcionalidad en APEINTADragon ya que esta aplicación funciona para varios idiomas, entre ellos el inglés.

Por si Dragon llegara a implementar el cálculo de estos índices para transcripciones hechas con perfiles en español en próximas versiones del SDK se ha optado por explicar sus posibles usos y ventajas. Esta funcionalidad se concibió como una ayuda para facilitar la identificación de los errores, aquellas palabras con índices por debajo de un determinado umbral son mostradas de diferente manera –resaltadas, coloreadas, etc.- al resto, de modo que la persona correctora pueda identificarlas más

rápidamente. Al facilitar su identificación se reduce el tiempo empleado para este fin y junto con las frases alternativas reducir el tiempo del proceso entero de corrección.

En la interfaz de APEINTADragon se ha incluido un botón que permite calcular estos índices (Fig. 27).

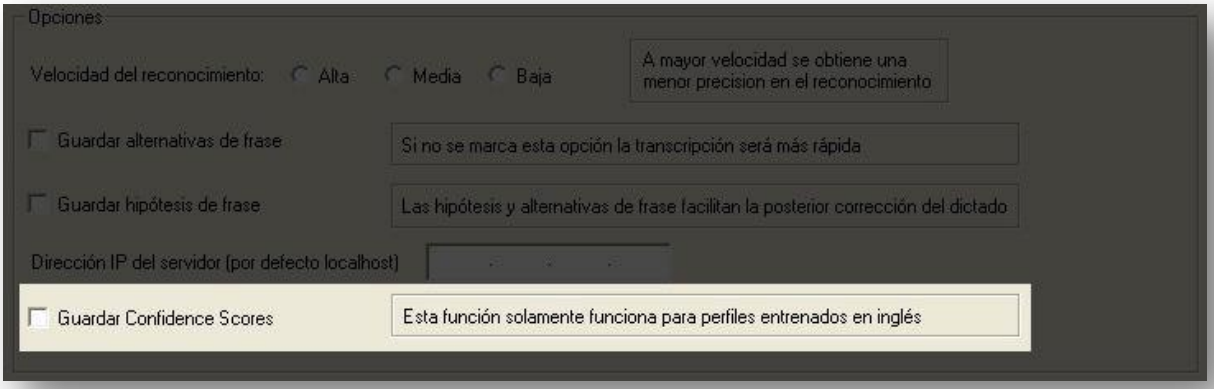


Figura 27. Se permite calcular los índices de confianza en la interfaz de APEINTADragon

Grabación del audio capturado

Otra funcionalidad que se ha incorporado es la grabación de la sesión en un archivo de audio. Éste se utiliza posteriormente durante la corrección para identificar, mediante su reproducción, las palabras erróneas contenidas en el texto. La escucha de la locución original es indispensable para corregir las transcripciones ya que permiten identificar los errores de palabra.

La captura y guardado del archivo de audio se realiza por medio del método *SnapshotSave*. Este método también genera un archivo *.IDX* que contiene únicamente el texto transcrito y los tiempos de inicio y fin de cada palabra en formato *xml*.

Reducción del retardo

Aunque esta funcionalidad no pretenda presentarse como una ayuda para el proceso de corrección de las transcripciones, sí que aborda otro problema que presenta el reconocimiento automático del habla: el retardo.

Durante las evaluaciones de APEINTA en escenarios donde se requería subtitulado en directo, uno de los problemas que se reportaron fue el retardo que se observaba desde que se pronunciaba una frase hasta que ésta era transcrita; en algunas situaciones llega a 30 segundos lo que puede provocar que se pierda el hilo del discurso y dificulte su comprensión.

Este retardo viene motivado porque Dragon utiliza el contexto, la totalidad de la frase, para ayudarse en el reconocimiento de una palabra y espera a que se produzca un silencio largo para identificar el final del discurso y construir la frase final –de

referencia. Para disminuir el retardo se debe aumentar la velocidad del reconocedor de voz lo que puede disminuir la precisión del mismo y aumentar el WER de la transcripción –a mayor velocidad menor precisión y viceversa.

En situaciones donde se necesita subtitulado en directo puede primar la velocidad con el fin de reducir el retardo, mientras que el subtitulado en diferido requiere de una mayor precisión en detrimento de la velocidad. Para poder adaptar el motor de reconocimiento de voz a las necesidades de cada situación se ha optado por añadir tres botones que modifiquen su velocidad (Fig. 28).

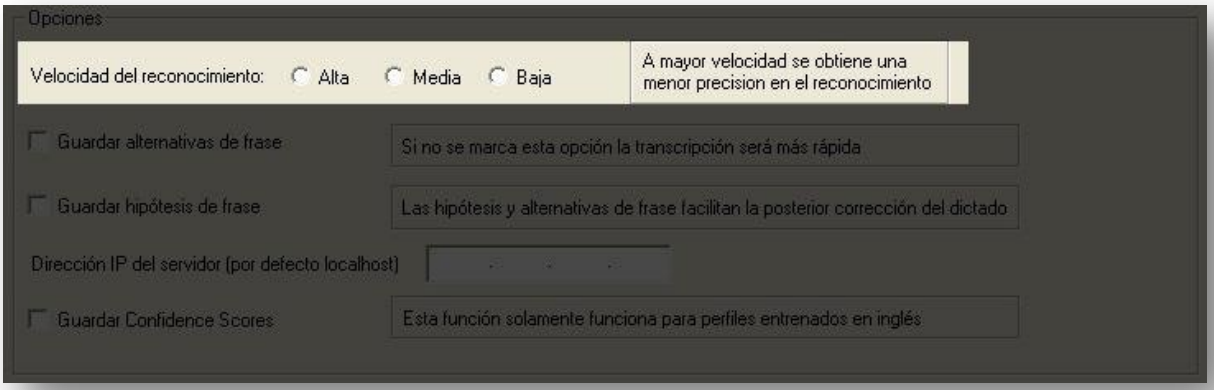


Figura 28. Opción para cambiar la velocidad del reconocimiento.

La diferencia entre los tiempos de reconocimiento de cada una de las tres velocidades es notable, por lo que esta funcionalidad reduce considerablemente el retardo.

Para poder modificar la velocidad del reconocimiento de voz es necesario modificar la constante *dgnengoptionComputeSpeed* de *DgnEngineOptionConstants*.

Hipótesis de palabra

Las hipótesis son aquellas palabras que se presentan como alternativas a las de referencia. El cálculo de estas palabras se realiza en el evento *phraseHypothesis* mediante el método *get_LastPhraseHypothesis,* que devuelve los resultados intermedios del reconocimiento antes de que se procese la frase completa. El proceso de reconocimiento comienza antes de que el locutor termine de pronunciar una frase, por lo que cada una de ellas generará múltiples hipótesis que pueden verse modificadas varias veces a lo largo del reconocimiento. La experiencia demuestra que además de verse modificado su contenido también crece el número de palabras que las forman a medida que avanza el reconocimiento de la frase.

Puesto que las hipótesis son resultados intermedios y se calculan antes de que se haya acabado de reconocer por completo la frase pronunciada, el acceso a las mismas puede hacerse antes que a las *frases alternativas* de la referencia –explicadas al

comienzo de este capítulo-; es decir, el evento *phraseHypothesis*, donde se calculan las hipótesis, tiene lugar justo antes que el evento *makeChanges*, donde se devuelven la frase de referencia y sus alternativas. Por este motivo la utilización de las hipótesis de palabra puede reducir el retardo del reconocimiento, lo que es especialmente ventajoso para las situaciones de subtitulado en directo.

```
<Phrase end="30485" id="2" start="28480">
  <RefWords numwrds="6">
    <WRD c="500" end="28819" id="1" srt="no" start="28640"/>
    <WRD c="500" end="29238" id="2" srt="entiendo" start="28819"/
    <WRD c="500" end="29418" id="3" srt="por" start="29238"/>
    <WRD c="500" end="29538" id="4" srt="qué" start="29418"/>
    <WRD c="500" end="29637" id="5" srt="lo" start="29538"/>
    <WRD c="500" end="30276" id="6" srt="menciona" start="29637"/
  </RefWords>
  <Hypothesis>
    <WRD c="0" end="2340" id="1" srt="no" start="1941"/>
    <WRD c="0" end="2460" id="1" srt="entiendo" start="2340"/>
    <WRD c="0" end="2579" id="1" srt="no" start="2460"/>
    <WRD c="0" end="3058" id="1" srt="entiendo" start="2579"/>
    <WRD c="0" end="2340" id="2" srt="por" start="1941"/>
    <WRD c="0" end="2460" id="2" srt="qué" start="2340"/>
    <WRD c="0" end="2579" id="2" srt="lo" start="2460"/>
    <WRD c="0" end="3577" id="2" srt="no" start="2579"/>
    <WRD c="0" end="3717" id="2" srt="entiendo" start="3577"/>
    <WRD c="0" end="28819" id="3" srt="por" start="28640"/>
    <WRD c="0" end="29238" id="3" srt="qué" start="28819"/>
    <WRD c="0" end="28819" id="4" srt="la" start="28640"/>
    <WRD c="0" end="29238" id="4" srt="mención" start="28819"/>
  </Hypothesis>
```

Figura 29. Ejemplo de las hipótesis generadas durante el reconocimiento de una frase.

En la Fig. 29 se puede ver como se han generado tres hipótesis durante el reconocimiento de la frase:

"No entiendo por qué lo menciona"

Cada una de las tres hipótesis de la frase anterior contiene un texto distinto y con mayor número de palabras que la anterior. Esto se produce por la evolución temporal y del reconocimiento de la frase pronunciada lo que genera diferentes frases.

"No entiendo"

"No entiendo por qué"

"No entiendo por qué la mención"

El problema que presentan las hipótesis es que los tiempos de inicio y fin de una misma palabra son distintos entre las diferentes hipótesis de una frase. Como se puede

ver en la figura anterior, la palabra “entiendo” tiene hasta tres tiempos diferentes, al igual que las demás palabras que coinciden en las tres hipótesis. La razón exacta que provoca este error no ha podido ser hallada durante la realización de este proyecto aun recurriendo a ayuda externa, por lo que este comportamiento se debe pues a la forma de operar del motor de reconocimiento del Dragon.

La principal consecuencia de este problema es que no se pueden utilizar los tiempos de inicio y fin de cada palabra para sincronizar el audio con la transcripción y, por tanto, recurrir a las hipótesis de palabra como un recurso para reducir el retardo existente.

Al igual que se ha hecho previamente, se ha incluido un control para poder calcular las hipótesis de palabra (Fig. 30).

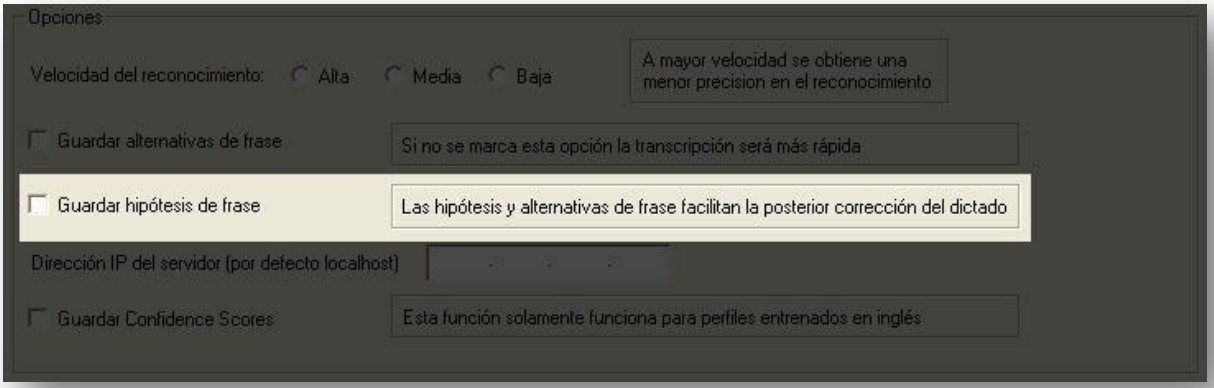


Figura 30. Alternativas de palabra o hipótesis.

Filtrado de sonidos

En ocasiones, se pronuncian sonidos no intencionados por el locutor o bien el micrófono capta sonido ambiente, y no se corresponden con ninguna palabra. Estos sonidos suelen transcribirse como palabras o letras sueltas que no tienen sentido en el contexto del discurso del locutor, lo que puede modificar el sentido de una frase y confundir al lector.

Para evitar las transcripciones de este tipo de sonidos se ha utilizado la constante *dgndictoptionCaptureAllDictation* de *IDgnDictationOptionConstants* en el método *initializeDictation()*.

Algunos de los sonidos que no constituyen palabras del discurso y que son pronunciados por el locutor suelen ser muletillas del tipo “eh”, “ah” o “mmhh”. En el caso de que estos sonidos deseen ser transcritos como muletillas, en casos donde se quiera recoger la totalidad del discurso, en lugar de “e”, “a” o “m” se debe utilizar la contante *dgnengoptionReturnPauseFillers* de *DgnEngineOptionConstants* en *initializeEngine()*. Esta constante solamente tiene efecto si la constante anterior

dgndictoptionCaptureAllDictation se ha activado previamente; *dgnengoptionReturnPauseFillers* detecta automáticamente si la constante *dgndictoptionCaptureAllDictation* está activada.

Signos de puntuación

Otro problema que fue notificado durante la evaluación de APEINTA en las asignaturas de Documentación y Biblioteconomía, es la ausencia de signos de puntuación en la transcripción. Esta ausencia no es trivial sino que se corresponde con un problema de difícil solución para cualquier paradigma del reconocimiento automático del habla. La no escritura de estos signos reside en la dificultad que supone para un reconocedor de voz identificar que pausas corresponden a uno u otro signo de puntuación. Aunque existen sistemas de reconocimiento del lenguaje natural que utilizan métodos más complejos y eficientes para puntuar las transcripciones, Dragon mide la longitud de los silencios para insertar un punto o una coma –no reconoce ningún signo más.

Sin embargo, las pruebas realizadas con el objetivo de evaluar esta funcionalidad demostraron que no es nada preciso. El sistema introdujo menos de una docena de signos en discurso de algo más de 45 minutos. Esto, según se indica en el API de Dragon, se debe a que la cadencia del discurso debe ser constante y las pausas algo más largas de lo normal para que la puntuación automática sea efectiva.

Para usar la puntuación automática se debe utilizar la constante *dgnitnoptionUseAutomaticPunctuation* de *DgnItnOptionConstants* en *initializeEngine()*.

Información de sincronismo entre audio y transcripción

Tras el proceso de reconocimiento Dragon puede generar un archivo *.DRA* que contiene el texto transcrito junto con información de tiempos de palabra, frases, índices de confianza, alternativas de frase, etc. Este archivo, con un tamaño de aproximadamente 1Mb por minuto de voz, es utilizado por el optimizador acústico para adaptar la información del perfil de voz. Como se explicó anteriormente adaptar el perfil de voz de un locutor acorde a sus características tonales mejora la precisión del reconocedor, por lo que se obtienen mejores resultados.

Al contener toda la información sincronizada relativa a la transcripción se observó que estos archivos podrían tener una función similar al *.XML* que se ha construido con el mismo contenido y asistir en el proceso de corrección. Sin embargo este tipo de archivos están codificados (Fig. 31) de forma que no son interpretables si no se utilizan aplicaciones que contengan el framework de Dragon.

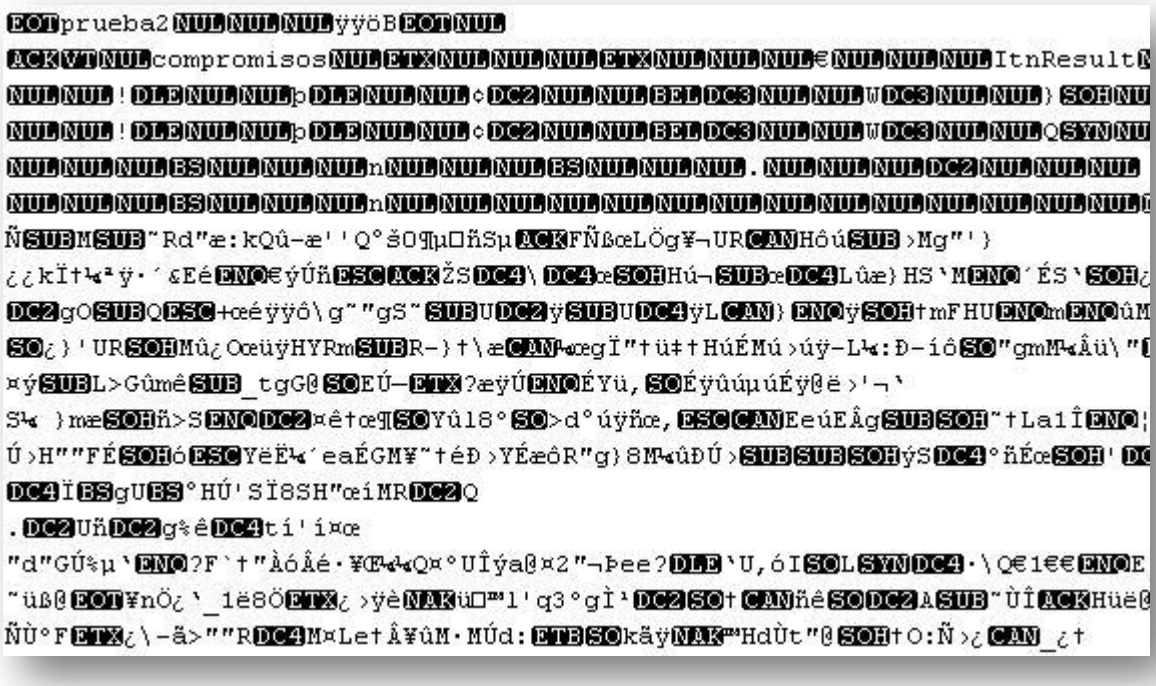


Figura 31. Extracto de un archivo *.DRA* generado por Dragon

Para obtener los archivos *.DRA* de cada una de las sesiones de voz se debe activar la constante *dgnengoptionCollectDictationData* de *DgnEngineControlConstants* en el método initializeEngine().

Dragon ofrece la posibilidad de convertir los archivos *.DRA* en archivos *.XML*, de formato estándar. Sin embargo, la herramienta que utiliza para convertir estos archivos está programada para mostrar únicamente la transcripción del texto, los índices de confianza de las palabras y los tiempos de inicio y fin de cada una de ellas; lo relativo a frases alternativas, hipótesis de palabra u otra información oculta no es convertida, aunque ésta existe en el archivo *.DRA* original. El documento *.XML* resultante de la conversión del archivo *.DRA* de la figura anterior se muestra en la Fig. 32.

```
<section>
  <text>
    <paragraph>
      <dgnspeech ID="SR_1" conf="500" dictationID="W_2" end="3757" start="3316">Serán </dgnspeech>
      <dgnspeech ID="SR_2" conf="500" dictationID="W_3" end="4158" start="3757">esos </dgnspeech>
      <dgnspeech ID="SR_3" conf="500" dictationID="W_4" end="4299" start="4158">los </dgnspeech>
      <dgnspeech ID="SR_4" conf="500" dictationID="W_5" end="4960" start="4299">compromisos </dgnspeech>
      <dgnspeech ID="SR_5" conf="500" dictationID="W_6" end="5181" start="4960">más </dgnspeech>
      <dgnspeech ID="SR_6" conf="500" dictationID="W_7" end="5601" start="5181">firmes </dgnspeech>
      <dgnspeech ID="SR_7" conf="500" dictationID="W_8" end="5702" start="5601">de </dgnspeech>
      <dgnspeech ID="SR_8" conf="500" dictationID="W_9" end="5782" start="5702">la </dgnspeech>
      <dgnspeech ID="SR_9" conf="500" dictationID="W_10" end="6544" start="5782">Unión Europea </dgnspeech>
    </paragraph>
  </text>
</section>
```

Figura 32. Archivo *.XML* resultado de la conversión de su homólogo en *.DRA*

Aunque un archivo *.DRA* tiene toda la información que se demanda para poder asistir en el proceso de corrección, el uso de estos archivos es limitado. A continuación se muestran las ventajas y desventajas de utilizar cada uno de los archivos mencionados.

Ventajas del DRA frente al XML:

- Toda la información está reunida en un mismo fichero, incluido el audio, que puede ser accedida mediante el API del Dragon. Permite fácilmente reproducir el audio correspondiente a una frase o a una palabra, sintetizar la voz de lo transcrito (TTS) y recuperar alternativas de frase y palabra.
- El XML generado por APEINTAdragon no contiene información del audio, que ha de obtenerse por otro lado, a través de la función *snapshotsave(),* para obtener una funcionalidad similar que el DRA –en este caso habría que implementar nuevas funciones que permitieran sincronizar el audio con la transcripción, mientras que con el DRA ya está hecho.

Desventajas del DRA frente al XML:

- La principal desventaja es que no se puede tener la información contenida en el DRA hasta que no ha terminado el reconocimiento total de cada frase, lo que significa que no se puede hacer uso de las hipótesis de palabra, entre otras funciones, y así reducir el retardo del reconocedor. Para generar un archivo DRA por cada frase reconocida hay que llamar a la función *sessionsave()* periódicamente lo que provoca un aumento de la carga de trabajo de la aplicación.
- La codificación del fichero es desconocida (Fig. 31). Existe la necesidad de utilizar el framework de Dragon para editar, lo que limita la portabilidad de la herramienta de edición a otros entornos –únicamente Visual C++ y VisualBasic- y además eleva el coste ya que es necesario disponer de la licencia de Dragon para utilizarlo.
- El XML es legible por un ser humano sin necesidad de utilizar ningún programa y además el XML permite una fácil exportación y conversión a distintos formatos. El XML puede ser manipulable de manera más sencilla con cualquier programa que se diseñe para tal propósito.

De este modo, algunas de las nuevas funcionalidades incorporadas a la aplicación APEINTAdragon se han utilizado para generar un documento XML que contenga no sólo la transcripción a texto del archivo de audio sino también alternativas de frase, hipótesis de palabra, tiempo de inicio y fin de cada frase y palabra de referencia e índices de confianza. La elección de almacenar toda la información relativa al reconocimiento en un fichero XML soluciona los problemas presentados por los archivos de sincronismo generados por Dragon.

5.2.2. CONCLUSIONES

En esta segunda parte del proyecto se pretendía modificar el sistema de RAH para optimizar el reconocimiento de la voz y facilitar la corrección de los errores cometidos. Se decidió estudiar el API facilitado por Dragon con el fin de encontrar funciones que permitieran modificar el reconocedor y obtener nuevos datos relacionados con el reconocimiento. Tras analizar las librerías se descubrió una serie de funcionalidades que podrían si no ayudar en el proceso de corrección de las transcripciones sí arrojar información más precisa sobre las mismas. Para ello se modificó la aplicación APEINTAdragon para que obtuviera:

- Alternativas a la frase de referencia.
- Tiempos de inicio y fin por palabra.
- Índices de confianza de cada palabra.
- Audio grabado de la sesión.
- Archivos de sincronismo (DRA) y XML.

Junto a esta nueva información, la nueva aplicación también se programó para que implementara otras funcionalidades de utilidad para solventar otros problemas relacionados con el reconocimiento automático de la voz:

- Reducción del retardo durante la etapa de reconocimiento.
- Hipótesis de palabra.
- Filtrado de sonidos cuya transcripción dificultan la comprensión del texto.
- Signos de puntuación.

Algunas de las nuevas incorporaciones no tuvieron el efecto deseado, como los signos de puntuación o las hipótesis de palabra. En el primer caso, el reconocedor de voz no identificó las pausas entre palabras que se correspondían con un signo de puntuación. Según indica Dragon en su documentación, si esto ocurriese se debe alargar o reducir las pausas entre palabras cuando se quiera introducir una coma o punto; tras las pruebas realizadas, la variación del tiempo de las pausas no significó mejora alguna. Por otro lado, el uso de las hipótesis de palabra se ve limitado por el problema con el tiempo de inicio de cada una de ellas, lo que dificulta utilizar estas palabras como alternativa para reducir el retardo introducido durante el reconocimiento; estas palabras se obtienen antes de que el reconocimiento haya concluido y por ello, antes que la frase de referencia final.

Y otras aunque de gran utilidad quedaban mermadas por las limitaciones impuestas por Dragon, como los índices de confianza, exclusivas para perfiles en inglés, o los archivos de sincronismo (DRA), dependientes del framework de Dragon para poder ser usados.

Algunas funcionalidades como las alternativas de frase, tiempos de palabra e hipótesis de palabra se utilizaron para construir un archivo XML que contuviera más

información del reconocimiento que la mera transcripción y sirviera de alternativa al DRA en situaciones donde su utilización no es recomendable.

En un escenario donde se requiere subtitulado en directo, la utilización de archivos XML puede resultar de más utilidad que un DRA. Una ventaja de los archivos XML frente a los dra es su menor tamaño, ya que a diferencia del dra éstos no incluyen el audio de la transcripción embebido. En estos casos no hay tanta necesidad de tener una copia del audio que se está capturando pues la corrección se hace en tiempo real y la persona que corrija puede recordar unos segundos de la misma; hay que recordar que escuchar la grabación del audio es indispensable para poder identificar los errores de palabra en casos de corrección en diferido. Sin embargo, el archivo XML sí que puede ser una buena asistencia en la corrección en directo de los subtítulos puesto que estos archivos incorporan los índices de confianza de cada palabra y éstos pueden ayudar a identificar las posibles palabras mal transcritas.

Si de lo contrario, el escenario precisa de subtitulado automático en diferido y la corrección de la transcripción no debe ser en tiempo real, la utilización de los archivos DRA puede la alternativa más útil. La utilización de archivos DRA y XML en un escenario con estas características será evaluada en el siguiente apartado de pruebas y resultados.

6. PRUEBAS Y RESULTADOS

Con el fin de evaluar la nueva información complementaria a la transcripción y mostrar el funcionamiento de los archivos xml y dra generados con la nueva versión de APEINTAdragon, se ha creado un prototipo de editor. No se ha prestado atención a la interfaz de usuario ni a la usabilidad ya que este editor no pretende ser una aplicación de corrección de texto real ni es el objetivo de este proyecto. A continuación se muestra el funcionamiento de la aplicación, a la que nos referiremos como Editor, que se ha preparado para que admita la lectura de ficheros de *.XML* y *.DRA*.

En primer lugar (Fig. 33), al ejecutar la aplicación, ésta pide abrir un perfil de voz para poder activar la opción de corregir por voz. Para completar la asistencia en el proceso de edición se ha añadido la posibilidad de corregir los errores de la transcripción por voz.

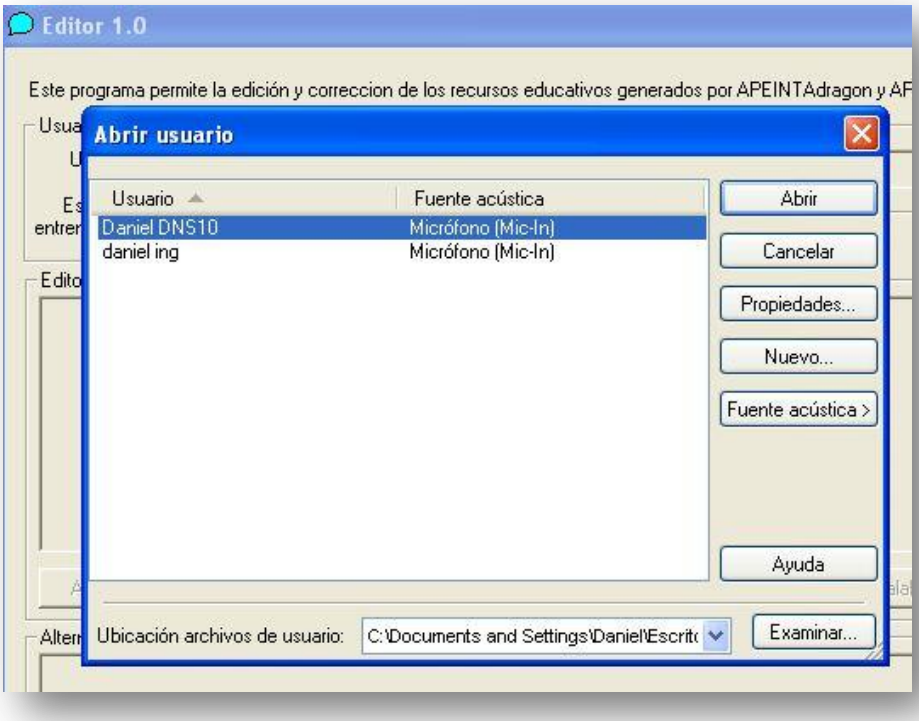


Figura 33. El Editor pide cargar un perfil de usuario para poder corregir la transcripción por voz.

Abriendo uno de los perfiles mostrados o bien entrenando uno nuevo si el corrector no tuviera ningún perfil asignado (presionando en el botón *Nuevo…*), la aplicación carga el perfil de voz y muestra la interfaz principal del programa (Fig. 34).

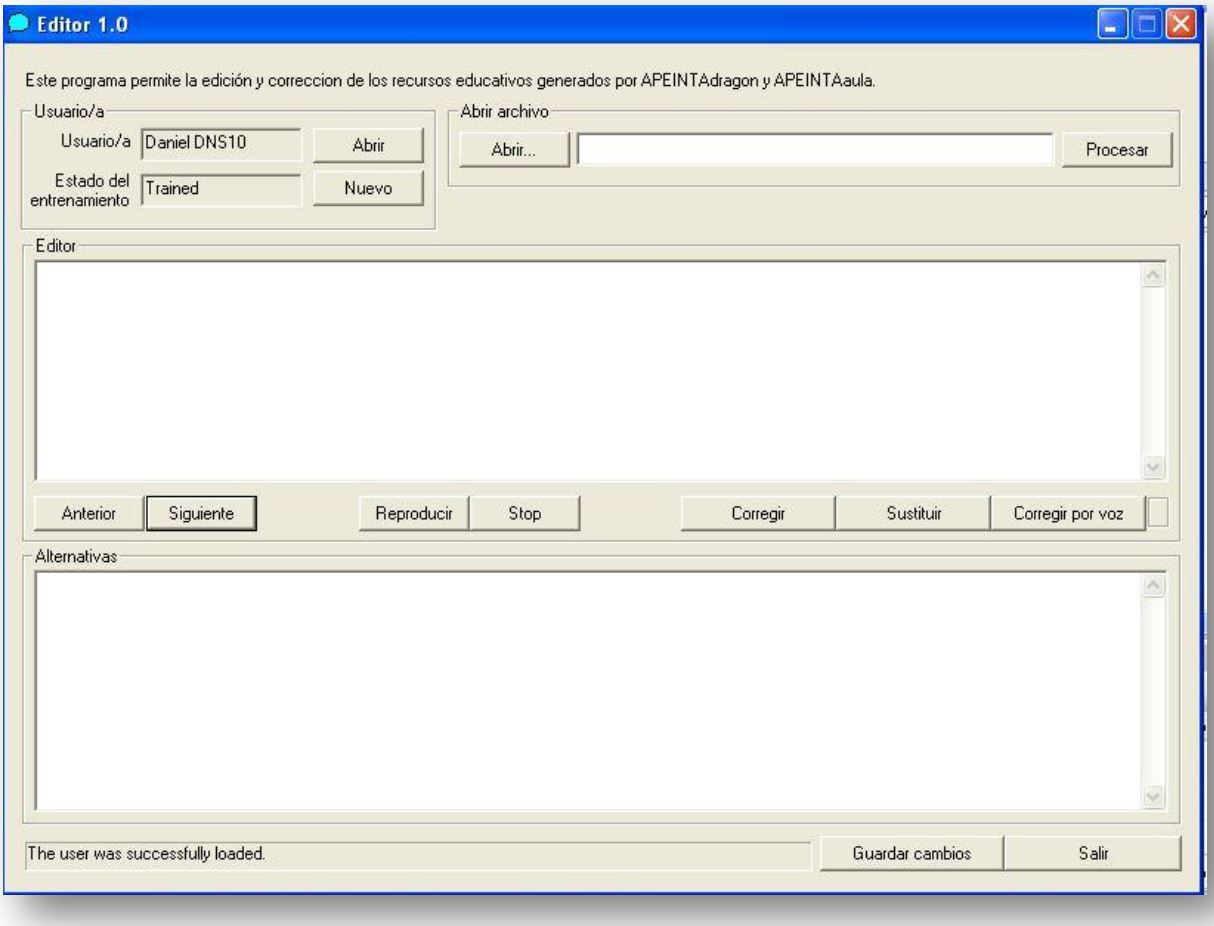


Figura 34. Interfaz principal del Editor.

En la interfaz del Editor (Fig. 34) se distinguen las siguientes zonas:

1. Usuario/a: Se muestra el nombre y el estado del entrenamiento (entrenado o sin entrenar) del perfil cargado en ese instante.
2. Abrir archivo: Este panel permite cargar el archivo xml o dra para que sea corregido.
3. Editor: Este campo de texto muestra la transcripción completa o por frases que contiene el archivo XML o DRA cargado.
4. Cuadro de botones: Estos botones permiten recorrer las frases de la transcripción, reproducir el audio de la transcripción y efectuar las correcciones pertinentes.
    - Anterior/Siguiente: Muestran las diferentes frases de las que está compuesta la transcripción. Habilitados únicamente para archivos XML.
    - Reproducir/Stop: Reproduce el audio sincronizado con la transcripción.
    - Corregir: Obtiene la palabra errónea que va a ser corregida.
    - Sustituir: Sustituye la palabra errónea por su alternativa.

- Corregir por voz: Enciende el micrófono y permite corregir por medio de la voz las palabras erróneas.

5. Alternativas: Este cuadro de texto muestra las alternativas existentes a la transcripción de referencia.
6. Guardar cambios y Salir: Los cambios realizados en la transcripción se guardan en el archivo xml o dra y se sale de la aplicación.

Corrección de archivos xml

Se debe cargar un archivo XML generado por la aplicación APEINTAdragon y presionar el botón *Procesar* para que se muestre en el cuadro de texto *Editor* (Fig. 35).

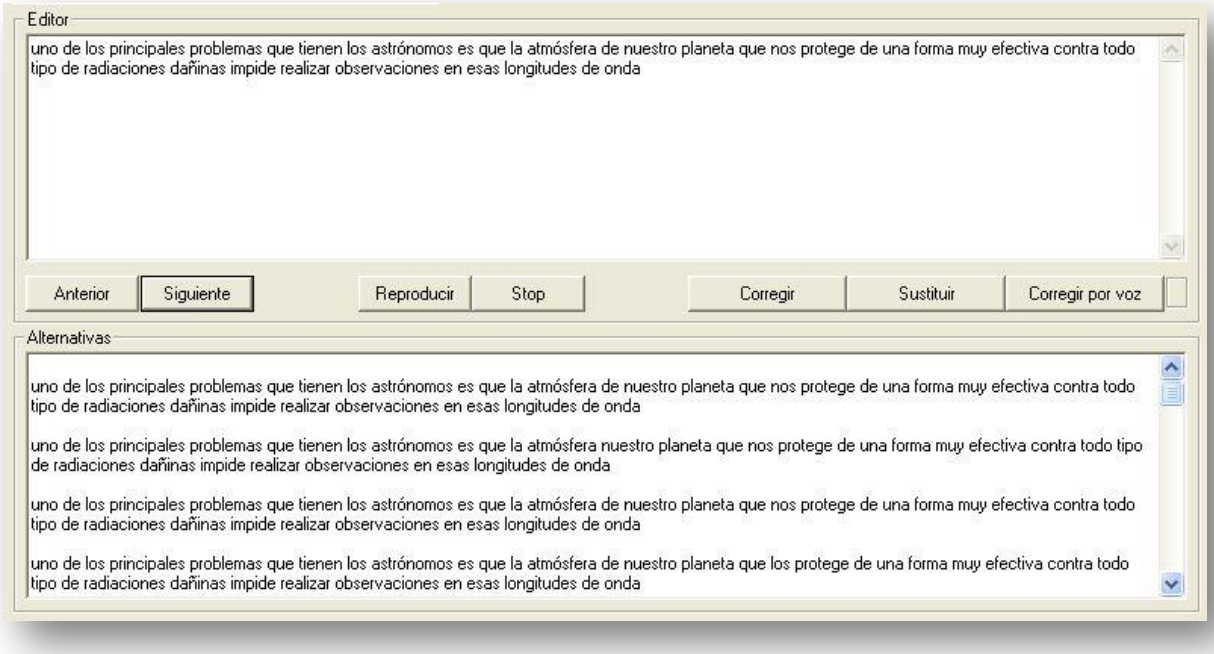


Figura 35. Transcripción y alternativas a esta de un documento xml.

En la Fig. 35 se observa la primera frase de las tres que componen la transcripción en este ejemplo así como en el cuadro de texto *Alternativas* se muestran las alternativas de la frase de referencia mostrada arriba. Los botones *Anterior* y *Siguiente* navegan y muestran las frases y sus alternativas que contiene el documento (Fig. 36). En el caso de que una frase no tuviera alternativas el cuadro de texto *Alternativas* mostrará el mensaje “NO EXISTEN ALTERNATIVAS PARA ESTA FRASE”.

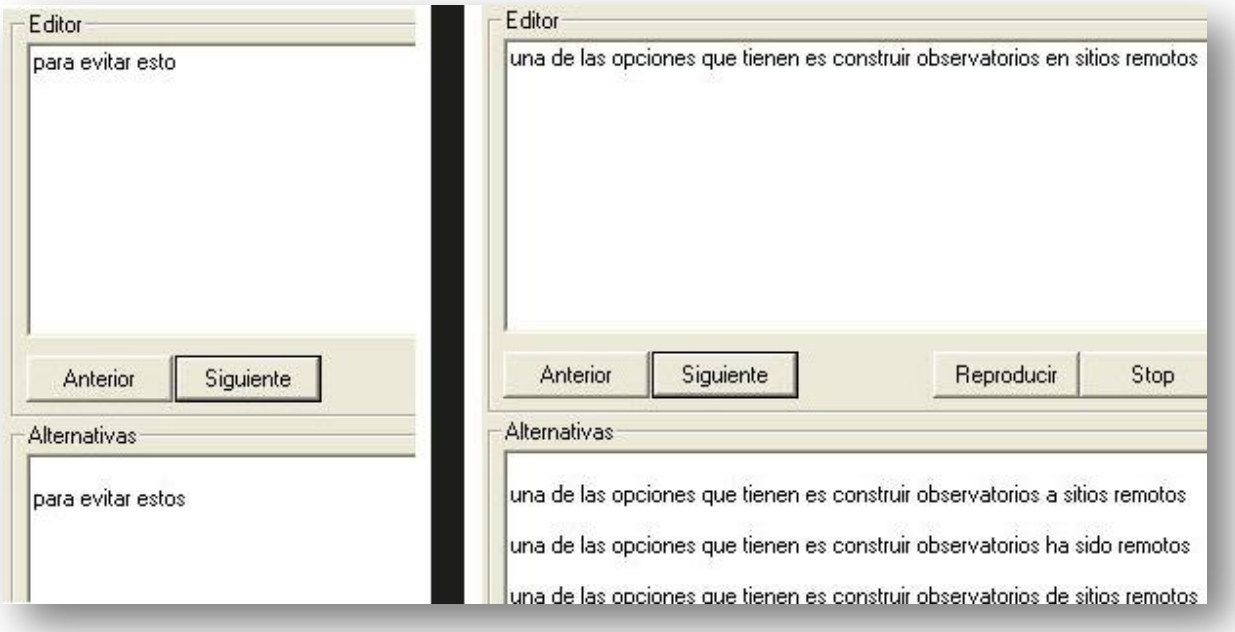


Figura 36. Diferentes frases de la transcripción que contiene el documento xml cargado.

Para realizar la corrección de una palabra errónea ésta debe se debe seleccionar. Se corregirá bien eligiendo alguna palabra de las alternativas mostradas, por voz –diciendo la palabra correcta- o simplemente escribiendo la nueva palabra. Si se optase por corregir la palabra sustituyéndola por otra de las alternativas ofrecidas, se deberá seleccionar la palabra alternativa y presionar el botón *Corregir palabra*. Si se procediera a corregir mediante voz, la nueva palabra expresada sustituirá a la palabra errónea seleccionada.

Corrección de archivos DRA

Al igual que en el caso anterior, se debe cargar el archivo DRA correspondiente y pulsar el botón *Procesar* para que la transcripción se muestre en el cuadro de texto *Editor*. Dragon muestra un cuadro de diálogo propio con las alternativas a la palabra o palabras seleccionadas para corregir (Fig. 37).

Figura 37.Cuadro de diálogo que se muestra al corregir una palabra utilizando un archivo DRA.

Como se ve en la Fig. 37. Dragon muestra una lista con las alternativas a la palabra seleccionada, de modo que para corregir la palabra errónea se debe seleccionar alguna de las éstas. Una de las ventajas de los archivos DRA es que permiten reproducir el texto seleccionado para corregir, facilitando así la identificación de la palabra correcta. Al igual que en el caso del archivo xml, la corrección se puede hacer por voz, sin necesidad de recurrir a las alternativas, si estas no fueran necesarias.

A pesar de no ser una aplicación óptima para este cometido –que no es su finalidad- este editor ha demostrado el potencial que puede tener la información complementaria obtenida y cómo puede servir de ayuda para reducir los tiempos de corrección en los diferentes escenarios que se puedan plantear.

7. PRESUPUESTO

En este capítulo se muestra la planificación inicial del proyecto y se listan los costes asociados a los medios materiales y a los recursos humanos utilizados en el desarrollo de este proyecto.

7.1. PLANIFICACIÓN INICIAL DEL PROYECTO

En la Fig. 38 –página siguiente- se muestra el diagrama de Gantt, donde se puede observar el desglose temporal y la duración asignada a cada tarea. La duración del proyecto es 11 meses naturales, con una jornada de 4 horas por día y con un periodo vacacional de 19 días en Agosto.

Los tiempos designados a cada tarea se han cumplido en su totalidad puesto que no han surgido inconvenientes durante la realización del proyecto.

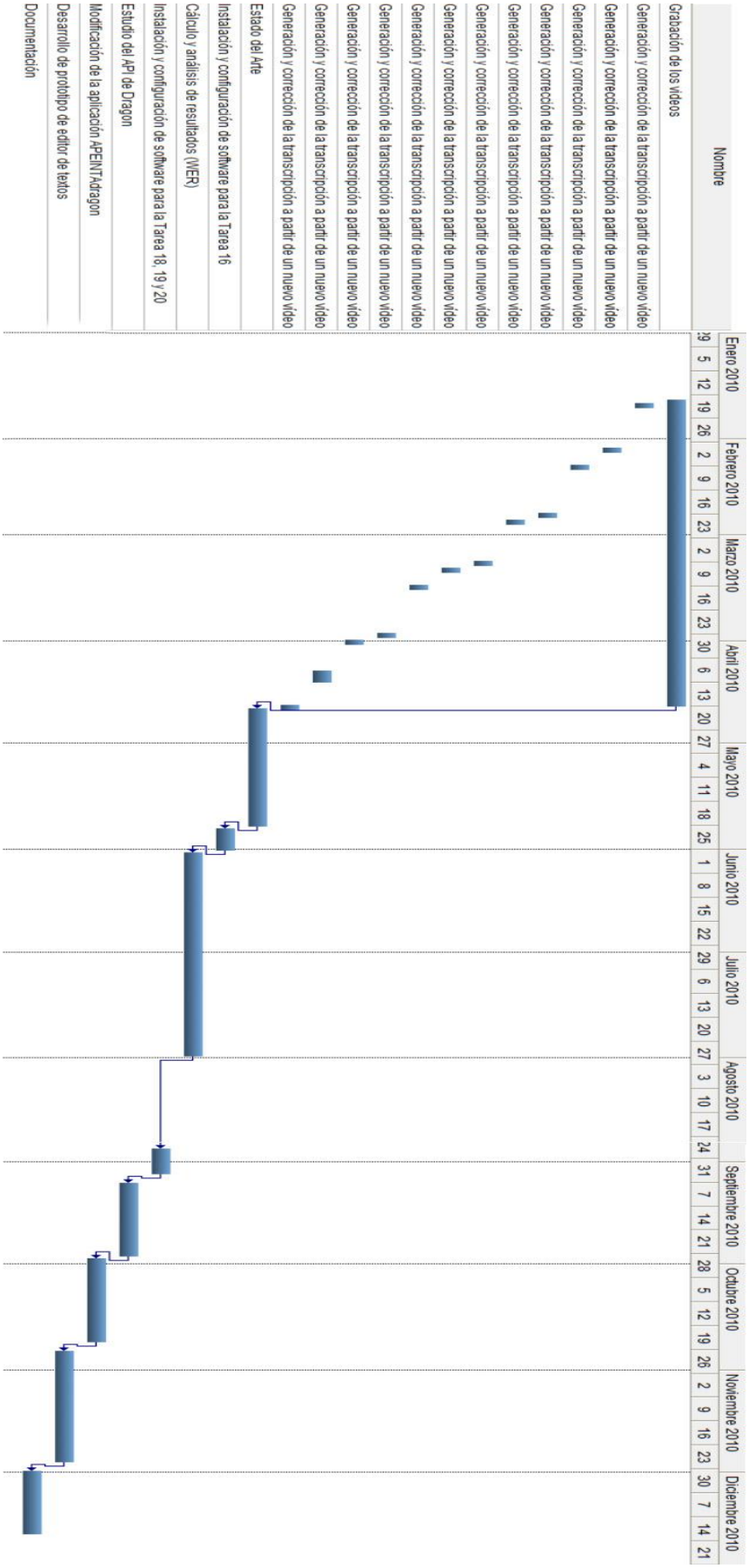


Figura 38. Diagrama de Gantt

7.2. RECURSOS HUMANOS

El coste en recursos humanos se refiere a los costes asociados al trabajo de las personas implicadas en el proyecto, que han sido:

- Un analista para las fases de análisis y diseño.
- Un programador que trabajan durante la fase de implementación y pruebas.

Los costes derivados de los recursos humanos aparecen en la Fig. 39:

Concepto	€/Hora	Horas	Coste (€)
Analista	14	240	3,360
Programador	9	676	6,084
Total			9,444

Figura 39. Costes derivados de los recursos humanos.

FUENTE: Colegio Oficial de Ingenieros Técnicos de Telecomunicación (COITT).

7.3. EQUIPAMIENTO

El hardware utilizado para la realización de este proyecto ha sido:

- Ordenador de sobremesa:
  - Sistema operativo: Windows XP Service Pack 3.
  - Sistema operativo: Linux (Kubuntu)
  - Características técnicas:
    - Procesador: Intel Core 2 6600
    - Memoria: 2Gb Ram

El software utilizado para el desarrollo de las aplicaciones implementadas y para el análisis de datos y resultados ha sido:

- Dragon NaturallySpeaking 9.5 SDK Client.
- Speech Recognition Scoring Toolkit (SCTK) Version 2.4.0.
- Aegisub Advanced Subtitle Editor
- Microsoft Visual Studio 2008.
- Microsoft Office 2007.

El coste asociado a los equipos hardware y software se muestra en la Fig. 40:

Concepto	Precio	Fuente
Ordenador de sobremesa	600€	www.pcbox.com
Dragon NS 9.5 SDK Client	4200€	www.nuance.es
SCTK 2.4.0	0€*	www.itl.nist.gov/iad/mig/tools
Aegisub Advanced Subtitle Editor	0€*	www.aegisub.org
Visual Studio 2008	880€	Microsoft Store
Microsoft Office 2007	139,00 €	Microsoft Store
Total	**5,819€**	

Figura 40. Coste de materiales

(*) Descarga gratuita.

7.4. COSTE TOTAL DEL PROYECTO

El coste total de los costes asociados a los materiales y a los recursos humanos se muestra en la Fig. 41:

Concepto	Precio
Costes RRHH	9,444€
Costes Materiales	5,819€
Total	**15,263€**

Figura 41. Coste total del proyecto.

8. CONCLUSIONES

La evaluación del sistema de subtitulado en distintos escenarios, como la Universidad Carlos III de Madrid o el Colegio Tres Olivos de Madrid, reveló un problema común a todos los sistemas de reconocimiento: los errores de transcripción cometidos en el proceso de reconocimiento de la voz y que comprometen la comprensión del mensaje transcrito. Sin embargo, los sistemas de reconocimiento de voz incorporan herramientas para reducir la tasa de error; en el caso de Dragon NaturallySpeaking éste incorpora la técnica de reentrenamiento del sistema para mejorar la precisión de sus motores, adaptando a las características del hablante el modelo de lenguaje, acústico y el vocabulario.

Tomando estos dos datos como punto de partida, en este proyecto se propuso analizar la evolución del WER de las transcripciones generadas por un sistema de reconocimiento que fue reentrenado hasta tres veces.

A la vista de los resultados obtenidos se puede observar que el reentrenamiento, en este caso por enrollment, ayuda a reducir la tasa de errores cometidos durante el reconocimiento. Si bien el reentrenamiento mejora el WER, la elección del material audiovisual para realizarlo condiciona notablemente la calidad del nuevo perfil generado. Para optimizar la eficiencia de este proceso la transcripción del archivo de audio debe estar libre de errores, por lo que se requiere corregir la transcripción literal. El archivo de audio debe tener una buena relación señal a ruido que permita al reconocedor de voz discriminar correctamente cada una de las palabras pronunciadas; además de ser recomendable unas características de sonido similares al resto de archivos de audio utilizados para reentrenar.

Por otro lado, el reentrenamiento prueba que la utilización de sistemas de subtitulado automático como APEINTA es más efectiva que realizar el subtitulado del material audiovisual desde cero, tanto por el tiempo como por los recursos invertidos. Siendo el tiempo de corrección/creación un aspecto a tener en cuenta, en el segundo caso no existe posibilidad de reducir drásticamente el tiempo invertido, mientras que en el caso de la generación automática de subtítulos sí puede verse reducido. A medida que el reentrenamiento mejora la precisión del sistema reconocedor, éste introduce menos errores y, por lo tanto, el tiempo de corrección disminuye progresivamente.

Dado que la metodología utilizada en la corrección de los errores de las transcripciones consume mucho tiempo, la segunda parte del proyecto se ha enfocado en encontrar soluciones que asistan en la tarea de corrección de los errores.

A través de la aplicación APEINTAdragon se ha modificado el reconocedor para obtener información adicional para apoyar y facilitar la corrección de las transcripciones. Entre otras funcionalidades incorporadas se han obtenido frases alternativas, índices de confianza, aumento de la velocidad del reconocimiento, etc. que

han mejorado el proceso de corrección de errores como otros aspectos problemáticos del reconocimiento de la voz.

Finalmente, este proyecto ha cumplido con los dos objetivos propuestos y explicados y además permite que, a raíz de esta investigación, se puedan realizar nuevos desarrollos que aprovechen el conocimiento obtenido en ambos.

9. TRABAJOS FUTUROS

En este capítulo se expondrán algunas propuestas para el desarrollo de futuros proyectos que pueden realizarse aprovechando la información alternativa a la transcripción obtenida con la aplicación APEINTAdragon.

En este proyecto se ha demostrado como la utilización de información complementaria a la transcripción, como frases alternativas, índices de confianza o hipótesis de palabra pueden resultar de utilidad en el proceso de corrección de las transcripciones producto del reconocimiento de la voz. Es por ello, que la principal línea de investigación futura es la adaptación y mejora de la interfaz del prototipo de editor de las transcripciones que incorpore todas las funcionalidades obtenidas en este proyecto.

Dado que la aplicación de subtitulado automático APEINTAdragon está pensada para funcionar tanto en directo como en diferido, la información complementaria, los archivos XML y las aplicaciones de edición utilizadas en la corrección de las transcripciones deberán estar adaptadas a cada escenario para maximizar su potencial, ya que los requerimientos para la corrección de las transcripciones en el subtitulado automático en directo o diferido son distintos.

En el primer caso, en el subtitulado en directo, el editor de las transcripciones debería estar preparado para realizar la corrección de los subtítulos en tiempo real. Para ello debe tener una interfaz optimizada para tal efecto y hacer uso de un documento XML que contenga la transcripción y la información complementaria obtenida para este escenario. Así es que, por ejemplo, el cálculo de las frases alternativas no solo no tiene tanta importancia como los índices de confianza que se pueden utilizar para identificar los errores rápidamente, sino que si éstas no se calculan aumentará la velocidad del reconocimiento y se reducirá el retardo, tan importante en estos casos.

En el caso de que la corrección de la transcripción no se haga en tiempo real, sino en diferido, el editor deberá ser configurado para poder utilizar el mayor número de alternativas de corrección y reducir el tiempo empleado para este fin.

En ambos casos, el editor debería tener una interfaz sencilla, que pueda ser utilizada sin apenas entrenamiento de la persona correctora. Para mejorar los tiempos y la calidad de la corrección sería bueno permitir que la corrección de una misma transcripción se realizase por dos o más personas al mismo tiempo, además, de que la interfaz permitiera realizar toda la corrección sin ayuda del ratón, tan solo con el teclado. Por otro lado la incorporación de herramientas de fraccionado automático de frases y analizadores sintácticos, entre otras, para la inclusión de signos de puntuación. Otra herramienta que podría ser implementada sería para la predicción de palabras en frases para la corrección automática de errores de palabra. Herramientas como

STILUS[29], permiten incorporar sistemas procesado lingüístico a aplicaciones basadas en tecnología del habla.

[29] http://www.daedalus.es/productos/stilus.html

10. GLOSARIO

RAH – Reconocimiento Automático del Habla.

WER – Word Error Rate.

SRT – Formato de texto utilizado en el subtitulado.

WAV – WAVeform audio file format.

XML – eXtensible Markup Language.

11. REFERENCIAS BIBLIOGRÁFICAS

[1] Ministerio de Educación y Ciencia, Coordinador: Francisco Jesús García Ponce, "Informe del Ministerio de Educación y Ciencia sobre Accesibilidad, educación y tecnologías de la información y comunicación", CNICE, Serie Informes Vol. 17, 2006.

[2] APEINTA [Página Web]
http://www.cesya.es/es/investigacion/trabajos/proyecto01

[3] Javier Jiménez, Pablo Revuelta, Ana Iglesias, Lourdes Moreno, (2010). Evaluating the Use of ASR and TTS Technologies in the Classroom: the APEINTA Project, June, 2010, ED-MEDIA 2010-World Conference on Educational Multimedia, Hypermedia & Telecommunications, AACE, ISBN: 1-880094-81-9, pp. 3976-3980.

[4] Speech Recognition Scoring Toolkit (SCTK) [Página Web]
ftp://jaguar.ncsl.nist.gov/current_docs/sctk/doc/sclite.htm

[5] Dragon NaturallySpeaking [Página Web]
http://www.nuance.com/dragon/index.htm

[6] Nuance [Página Web]
http://www.nuance.com/

[7] ViaVoice [Página Web]
http://www.liberatedlearning.com/technology/index.shtml

[8] Tomáš Beran, Vladimír Bergl, Radek Hampl, Pavel Krbec, Jan Šedivý, Bořivoj Tydlitát and Josef Vopička, "Embedded ViaVoice", TSD 2004, LNAI3206, 2004, pp. 269-274.

[9] Hidden Markov Model Toolkit [Página Web]
http://htk.eng.cam.ac.uk/

HTK Book
http://www.spsc.tugraz.at/courses/scl/download/htkbook.pdf

[10] Departamento de Ingeniería de la Universidad de Cambridge [Página Web]
http://www.eng.cam.ac.uk/

[11] CMUSphinx [Página Web]
http://cmusphinx.sourceforge.net/

[12] Universidad Carnegie Mellon [Página Web]
http://www.speech.cs.cmu.edu/

[13] Keith Vertanen, "Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments", 2006, Technical Report, Cavendish Laboratory, University of Cambridge.

[14] Yunjia Li, Mike Wald, Shakeel Khoja, Gary Wills, David Millard, Jiri Kajaba, Priyanka Singh, Lester Gilbert, "Synote: Enhancing Multimedia E-Learning with Synchronised Annotation", in Proc. of the first ACM international workshop on Multimedia technologies for distance learning, ACM. ISBN 978-1-60558-757-8, 2009, Learning Societies Lab, Department of Electronics and Computer Science, University of Southampton, pp. 9-18.

[15] Instituto Tecnológico de Massachusets (CSAIL) [Página Web] http://www.csail.mit.edu/

[16] James R. Glass, Timothy J. Hazen, D. Scott Cyphers, Ken Schutte and Alex Park, "The MIT Spoken Lecture Processing Project", in Proc. of HLT/EMNLP 2005 Demonstration Abstracts, Vancouver, Octubre 2005, pp. 28–29.

[17] Caption Editing System [Página Web] http://www.alphaworks.ibm.com/tech/ces

[18] Instituto Nacional de Estadística (INE), "La discapacidad auditiva y visual en España: Encuesta sobre discapacidades 2008", Madrid, 2008.

[19] Wald, M., "Using Automatic Speech Recognition to Enhance Education for All Students: Turning a Vision into Reality" in Proc. Of 34th ASEE/IEEE Frontiers in Education Conference, Octubre 19-22 2005, Indianapolis, Indiana, ISBN: 0-7803-9078-4, pp. 22-25.

[20] C. García, D. Tapias, "La frecuencia fundamental de la voz y sus efectos en reconocimiento de habla continua", División de Tecnología del Habla Telefónica Investigación y Desarrollo, Revista de Procesamiento de Lenguaje Natural, Vol. 26, Septiembre 2000, pp. 163-168.

[21] Dynamic Programming Algorithm [Página Web] ftp://jaguar.ncsl.nist.gov/current_docs/sctk/doc/sclite.htm#dynam_prog_0